

pom's

La revue francophone des utilisateurs de l'Apple

Disque virtuel 64 K

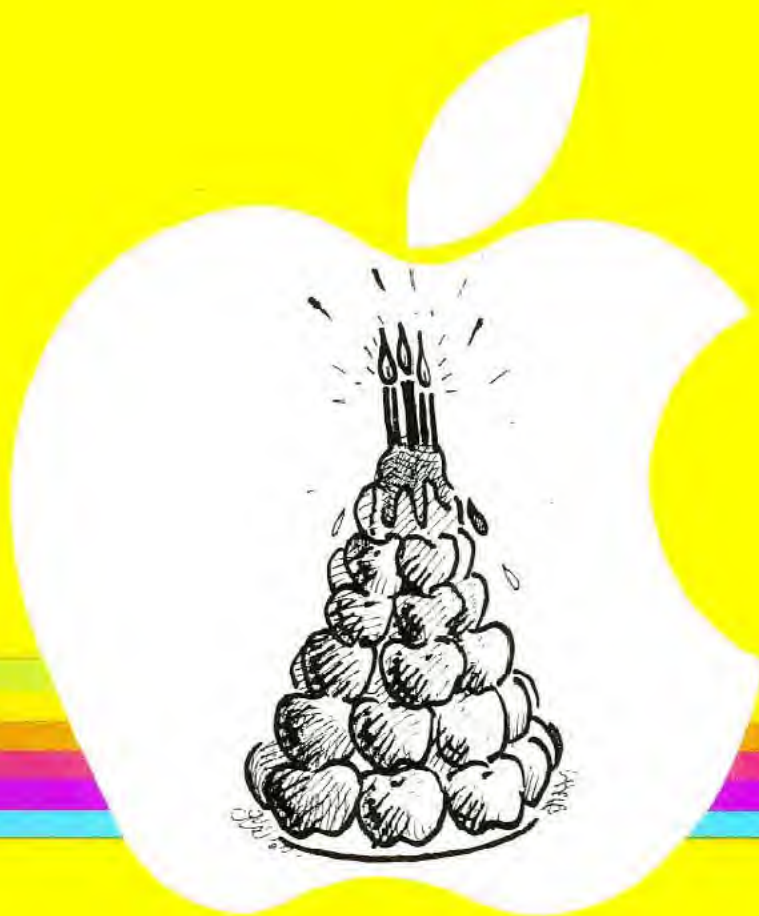
Amper-Interpréteur ICARE

Appleworks et Jane à l'essai

Un catalogue général

217 fichiers par disquette

Rubrique Macintosh



NUMERO 14 • PRIX 40 F

ISSN : 0294-0068

Souris créative cherche amitié passionnée pour fonder club.

Pour tous les passionnés de l'ordinateur personnel,
les branchés et ceux qui le sont moins, Apple lance le Club Apple.

Enfin un club avec des idées et des services
pour comprendre, pour gagner, pour s'évader.

Le Club Apple, c'est l'esprit Apple.

C'est le fruit de la passion.



PROMO 2000

Pour en savoir plus sur le Club Apple - le fruit de la passion - et connaître tous les avantages que nous réservons aux membres du club, découpez dès aujourd'hui le bon et retournez-le à Club Apple, avenue de l'Océanic - ZA de Courtabœuf - BP 131, Les Ulis cedex 91944. Vous recevrez sans engagement de votre part toutes les informations pour devenir membre du club.

Nom _____ Prénom _____

Adresse _____

Code postal _____

Sommaire

	Page	Langage *	Niveau **
Editorial par Hervé Thiriez	5		
Initiation à l'assembleur par Gérard Michel	7	B-A	T
Procédure & par Philippe Le Dru	15	A	M-T
Disque virtuel 64K par Jacques Rey	19	A	M-T
Appleworks à l'essai par Guy Lapautre	23		
Jane : L'Apple II sauce Macintosh par Jean-Michel Gourévitch	24		
Amper-interpréteur ICARE par Pascal Cantot	25	B-A	M
Visicalc Advanced Version par Hervé Thiriez	37		
Les essais du Macintosh par Hervé Thiriez	43		
De MacWrite à Apple Writer //e par Pom's	45	B	T
Les tokens du Basic Microsoft par Jean-Luc Bazanegue	46	(B)	T
Gestion de masques améliorée par Jacques Tran-Van et Maurice Schapiro	47	B-A	T
Lève-toi et brille par Alexandre Avrane	53		T
217 fichiers par disquette par Marcel Cottini	55	B	M
Conversion minuscules/MAJUSCULES par Alexandre Avrane	56	A	M
Un catalogue général par Michel Baudrand	57	B-A	T
Koala pad à l'essai par Marcel Cottini	67		
Micro-informations par Jean-Michel Gourévitch	69		
Lecture/écriture rapide de tableaux par Jacques Duma	73	B	T
INPUT généralisé miniature par Jacques Duma	73	B	T
Courrier des lecteurs par Alexandre Duback	75		

* Langage : B(asic) - A(sembleur) - P(ascal). (B) signifie : relatif au BASIC.

** Niveau : D(ébutant) - M(oyen) - P(rofessionnel) - T(ous).

P-T signifie : programme utilisable par les débutants, mais dont la compréhension est de niveau "Professionnel".

Les annonceurs

ACER MICRO : p. 18 / Apple : p. 38-39 / ASAP : p. 72 / B.F.I. : p. 79 / Club Apple : p. 2 / COMPUTER 3 : p. 74 / DYNAMITE COMPUTER : p. 77 / HELLO : p. 71 / HIT Microsystème : p. 80 / IEEE : p. 68 / INFORTEC : p. 68 / KARDEX : p. 13 / LIST : p. 51 / M.B.D.C. : p. 13 / MEMSOFT : p. 44 / L'Ordinateur Individuel : p. 6 / PEACHTREE : p. 40-41 / P.S.I. Diffusion : p. 42 / SATELCOM : p. 77 / SICOB : p. 4 / INFOSOFT : p. 66 / TELECOMPO : p. 74

Édition MEV — 64-70 rue des Chantiers — 78000 Versailles

Directeur de la publication : Hervé Thiriez, Imprimerie Rosay, 94300 Vincennes. Imprimé en France. Dépôt légal : 3^e trimestre 1984.



QUESTIONS D'AUJOURD'HUI?

REPONSES AU SICOB.

SICOB 84

***CNIT-PARIS LA DEFENSE
DU 22 AU 28 SEPTEMBRE
(SAUF DIMANCHE 23) DE 9 H A 18 H.
JOURNEES PROFESSIONNELLES: 19, 20, 21 SEPT.***

Trois ans déjà ! Trois années pendant lesquelles Pom's a augmenté sa pagination (ce numéro comporte 80 pages), sa fréquence de parution, son tirage, et même son prix de vente, quoique moins de 15 % d'augmentation en trois ans, cela soit un record en matière d'inflation ! Surtout avec le papier qui a pratiquement triplé et les frais postaux qui n'ont pas respecté les consignes du gouvernement en matière de contrôle des prix...

Trois articles de ce numéro valent chacun le prix d'un abonnement annuel : tout d'abord le catalogue général de **Michel Baudrand**, et le programme Icare de **Pascal Cantot** qui regroupe (tel Haïfa) un paquet de routines toutes plus intéressantes les unes que les autres, sous le contrôle du célèbre &. Puis, inspiré par l'article de Michel Haag paru dans le numéro 12, la transformation par **Jacques Rey** de votre carte 80 colonnes "étendue" en un disque virtuel de 64K.

Marcel Cottini stocke 217 fichiers sur une disquette. Il vous propose également un essai de Koala Pad. Dans le domaine des essais, vous trouverez également le logiciel Jane présenté par **Jean-Michel Gourévitch**, et Appleworks, logiciel intégré vu par **Guy Lapautre**. Pour ma part, j'ai pu tester Visicalc Advanced Version, le dernier développement de Visicalc.

Deux articles originaux d'**Alexandre Avrane** : d'une part une routine de conversion de l'affichage minuscule en majuscule pour que vos programmes conçus sur //e et //c soient utilisables sur Apple II sans ROM LC, et d'autre part, une contribution originale en Visicalc qui vous donnera les heures de lever et coucher du soleil.

Jacques Tran-Van et **Maurice Schapiro** ont modifié le programme de gestion de masque écrit par **Gérard Michel**. Ce dernier poursuit par ailleurs sa série d'initiation à l'assembleur pour laquelle nous recevons toujours un courrier enthousiaste.

Philippe Le Dru ajoute à l'Applesoft une routine de gestion de procédure avec des variables locales tout à fait "pascaliennes". Quant à **Jacques Duma**, il vous délivre des problèmes de virgules et deux-points dans les INPUTs en deux lignes d'Applesoft. En deux lignes également vous pourrez sauvegarder et recharger des tableaux de variables sur disquettes.

La rubrique Macintosh grandit et va devenir, probablement à partir du numéro de novembre, un cahier complet. Je vous y présente une analyse de deux livres et deux logiciels de jeu. **Jean-Luc Bazanegue** a recherché pour vous les tokens du Basic Microsoft du Macintosh, et **Pom's** vous donne le moyen d'envoyer à votre brave Apple // des textes écrits avec MacWrite. Vous êtes en effet de plus en plus nombreux à disposer d'un Mac, et Pom's tient à apporter son soutien aussi aux utilisateurs à ce nouveau bourreau des cœurs.

Nous avons encore de nombreux programmes passionnants en réserve et prions les auteurs qui nous envoient leurs contributions de bien vouloir prendre patience. Dans le prochain numéro (novembre), entre autres beautés, vous aurez "Mobby Disk", un programme qui permet d'analyser ou de modifier un disque à volonté.

Enfin, Pom's continue à rechercher un collaborateur connaissant parfaitement bien l'assembleur et le Basic Applesoft, sachant s'exprimer clairement et sans fautes d'orthographe, pour un travail de rédaction pouvant aller du mi-temps au temps complet.

Hervé Thiriez

Erratum : sur la couverture, il s'agissait de Bill **Budge** et non **Bodge**. Devons-nous y voir l'influence néfaste de la traduction de **bug** en **Bogue** ?

Ont collaboré à ce numéro : Alexandre Avrane - Michel Baudrand - Pascal Cantot - Marcel Cottini - Alexandre Duback - Jacques Duma - Jean Michel Gourévitch - Guy Lapautre - Philippe Le Dru - Gérard Michel - Jacques Rey - Maurice Schapiro - Hervé Thiriez - Jacques Tran Van. **Rédacteurs** : Alexandre Avrane - Gérard Michel. **Directeur de la publication, rédacteur en chef** : Hervé Thiriez. **Dessin** : Laurent Bidot.

Siege social et abonnements : Editions MEV - 64-70, rue des Chantiers - 78000 Versailles - Tél (1) 951.24.43

Rédaction et courrier des lecteurs : 59, bd de Glatigny - 78000 Versailles.

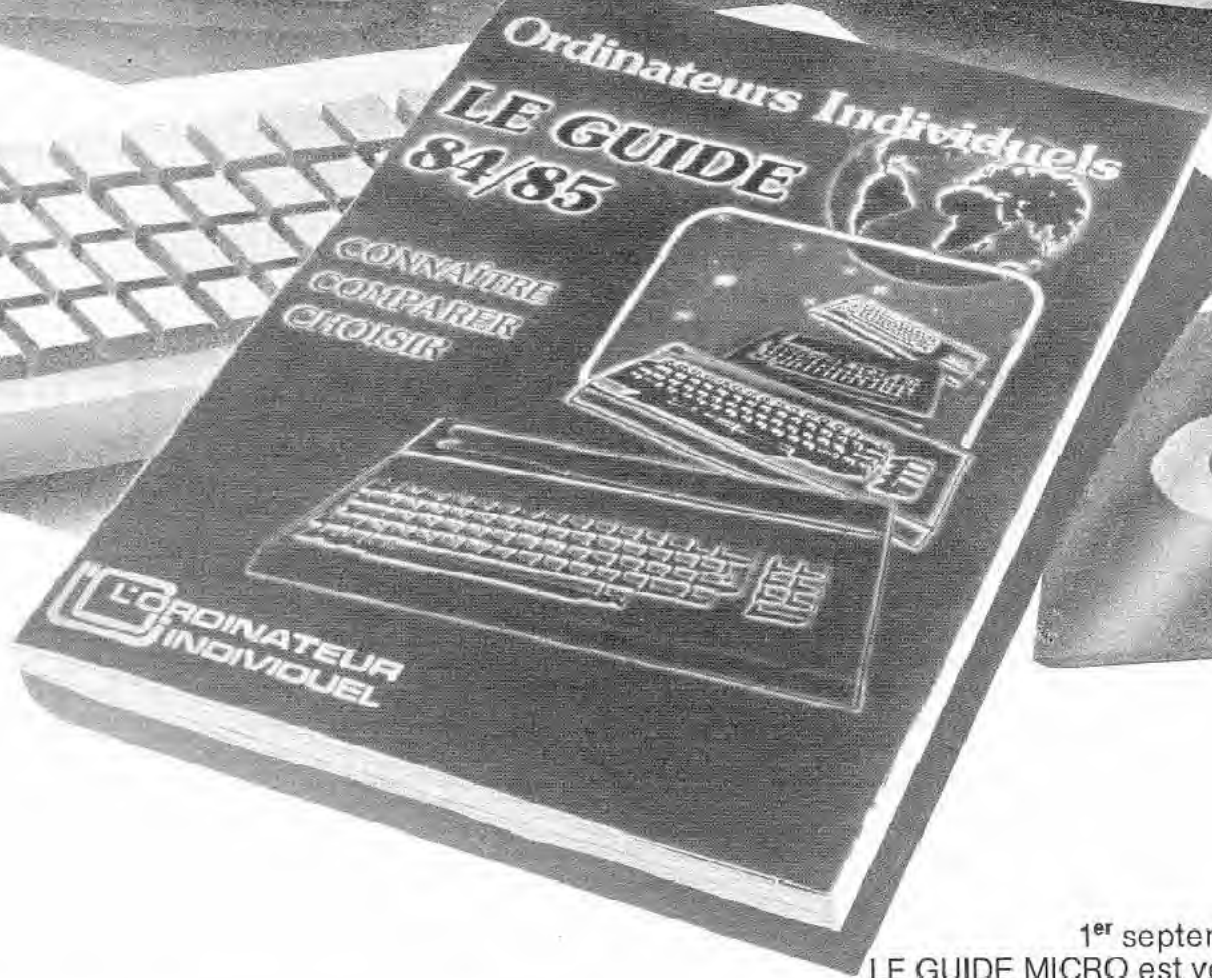
Règle publicitaire : Forcé 7 - Anne Jourdan - 5, place du Colonel Fabien - 75010 Paris - Tél 240.22.01

Diffusion N.M.P.P. : Sophie Mamez - Tél (1) 240.22.01

Composition : Télécompo - 13-15, avenue du Petit Parc - 94300 Vincennes - Tél 328.18.63

Impression : Rosay - 47, avenue de Paris - 94300 Vincennes - Tél 328.18.63

LA BIBLE



1^{er} septembre 1984 :
LE GUIDE MICRO est vendu 35 FF
chez votre marchand de journaux.

1980 : il était déjà là avant même l'explosion de la micro-informatique, prêt à évoluer avec elle d'année en année.

1984/1985 : il est "la" Bible. Son titre, ses fidèles le lui ont donné, naturellement : "LE" GUIDE.
Instrument de première approche des néophytes.
Référence des passionnés, des vrais amateurs, des professionnels, des constructeurs, des vendeurs.

1984/1985 : LE GUIDE MICRO présente :

- les 216 fiches techniques détaillées des micro-ordinateurs du marché (de 500 FF à 50 000 FF),
- le tableau exclusif de plus de 130 imprimantes,
- un panorama des 60 logiciels à connaître dans toutes les applications,
- près de 2000 adresses de constructeurs, de clubs et de boutiques, par ordre alphabétique et par département,
- les mots clefs de l'informatique.

1984/1985 : élaboré par une équipe de journalistes, LE GUIDE MICRO explique : qu'est-ce qu'un ordinateur ? Que permet-il ? LE GUIDE MICRO prend parti et dit pourquoi : pour ou contre 55 ordinateurs individuels. Il commente les grandes manœuvres de l'année 84. Il tutoie l'avenir : les ordinateurs de janvier 85 sont au rendez-vous.

LE GUIDE DE L'ORDINATEUR INDIVIDUEL

Initiation à l'assembleur (4)

Gérard Michel

L'article précédent nous a donné l'occasion de rentrer un peu dans le détail du contenu des octets manipulés par votre Apple et de voir comment certaines instructions de l'assembleur travaillent au niveau des bits d'information.

En analysant les traitements présentés au sujet des instructions ADC ou SBC, vous avez pu constater qu'il n'y a finalement rien de bien mystérieux dans l'évolution de ces suites de 1 et de 0 qui constituent les octets. L'application rigoureuse de règles arithmétiques simples permet en effet de prévoir le résultat des opérations binaires et, par voie de conséquence, de comprendre le fonctionnement et les conditions d'emploi des instructions du langage assembleur réalisant ces opérations.

Nous pouvons donc maintenant aborder d'autres instructions plus "fines" quant à leur champ d'application. Si, comme toujours, elles manipulent la quantité économique d'information qu'est l'octet, les conséquences de cette manipulation ne peuvent être totalement comprises qu'à l'aide d'un examen bit à bit. Des règles simples conduisent là encore à la prédiction du résultat, et c'est bien normal lorsque les objets du traitement sont aussi "pauvres" et peu nombreux que 0 et 1.

Les opérateurs logiques

Nous retrouvons là des instructions présentes également dans l'Applesoft. Lorsque vous écrivez "IF A=2 AND B=3" vous savez que la condition ne sera satisfaite que si l'on a à la fois "A=2" et "B=3". De même, si la condition devient "IF A=2 OR B=3", vous savez qu'il suffira que "A=2" ou que "B=3" pour qu'elle soit satisfaite, même si l'on n'a pas "B=3" lorsque "A=2", ou "A=2" lorsque "B=3".

Si l'on considère que "vrai" est codé par "1" et que "faux" est codé par "0", "A=2 AND B=3" ne sera vrai que si "A=2" donne 1 et si "B=3" donne 1. En revanche, pour que "A=2 OR B=3" soit vrai, il suffit que l'un des deux éléments "A=2" ou "B=3" donne 1. Des principes identiques sont utilisés pour les opérateurs logiques en assembleur, avec toutefois une distinction supplémentaire pour le "OR".

Le ET logique

Le code mnémorique de l'instruction est **AND**. Elle réalise un "ET logi-

que" entre l'accumulateur et ce qui suit le code **AND**, c'est-à-dire un nombre ou le contenu d'une case-mémoire désignée par l'un des modes d'adressage déjà étudiés. Le résultat est remis dans l'accumulateur.

Le ET s'effectue bit à bit, selon la table de vérité suivante :

0 AND 0 = 0

0 AND 1 = 0

1 AND 0 = 0

1 AND 1 = 1

Si le bit 3 de l'accumulateur est à 1 au moment du **AND** et que le bit 3 de l'élément "comparé" est à 0, on retrouvera le bit 3 de l'accumulateur à 0 à l'issue du **AND**.

Supposons ainsi que l'accumulateur contienne "01001101" (soit #\$4D) et l'adresse \$18 "11011000" (soit #\$D8). Après un **AND** avec le contenu de l'adresse \$18, on trouvera dans l'accumulateur la valeur "01001000" (soit #\$48) : sur le bit 0 (le plus à droite) on a 1 et 0 qui donnent 0, sur le bit 1 on a 0 et 0 qui donnent 0, sur le bit 2 on a 1 et 0 qui donnent 0, sur le bit 3 on a 1 et 1 qui donnent 1... et ainsi de suite.

Cette instruction **AND** permet en particulier de "forcer" des bits de l'accumulateur à 0. Prenons la valeur #\$EF, par exemple, soit en binaire "11101111". Un **AND** #\$EF forcera le bit 4 (le cinquième en partant de la droite) de l'accumulateur à 0 sans modifier les autres bits. Si l'accumulateur contenait "01110101", on aurait alors :

01110101 AND 11101111 → 01100101

Tous les bits de l'accumulateur demeurent inchangés, sauf le bit 4 qui passe à 0. Si ce bit 4 avait déjà été à 0 avant **AND**, il le serait resté après. En tout état de cause, on est certain de trouver 0 dans ce bit 4 après **AND**.

En assembleur, cela s'écrirait :
LDA #\$75
AND #\$EF
et l'accumulateur contient maintenant #\$65 (01100101).

Revenons un instant sur l'un des programmes d'illustration proposés dans l'article précédent. Nous avions vu qu'il fallait retirer # \$80 au code-clavier d'un chiffre pour obtenir son code-écran en mode INVERSE. Pour ce faire, nous avons utilisé SBC #\$80. Or les codes écran concernés

sont supérieurs à #\$80 et leur bit 7 (le plus à gauche) est donc à 1. En binaire, #\$80 vaut 10000000 et le complément à 2 de #\$80 vaut également 10000000 (calculez-le pour vous en convaincre). Retirer #\$80 d'un octet dont le bit 7 est à 1 revient donc finalement à mettre simplement son bit 7 à 0 (1 + 1 - 0) sans modifier ses autres bits. On pourrait obtenir le même résultat par un **AND** #\$7F (#\$7F = 01111111) qui force le bit 7 à 0 sans affecter les autres. Dans la mesure où l'on ne connaît pas toujours la valeur du bit C avant d'exécuter SBC, cette seconde solution peut nous permettre de faire l'économie d'une instruction SEC en garantissant un résultat équivalent. En résumé, si le bit 7 de l'accumulateur est à 1, **AND** #\$7F est équivalent à SEC suivi de SBC #\$80.

Les différents formats de l'instruction **AND** sont (ADS désignant une adresse symbolique) :

- **AND** nombre = **AND** #NN = 29 NN (par exemple 29 7F)
- **AND** adresse page 0 = **AND** ADS = **AND** \$aa = 25 aa (le ET est effectué entre l'accumulateur et le contenu de l'adresse aa)
- **AND** adresse quelconque = **AND** ADS = **AND** \$aaaa = 2D octet bas/octet haut (**AND** \$ED42 = 2D 42 ED)
- **AND** adresse page zéro, X = **AND** ADS,X = **AND** \$aa,X = 35 aa
- **AND** adresse quelconque, X = **AND** ADS,X = **AND** \$aaaa,X = 3D octet bas/octet haut
- **AND** adresse quelconque, Y = **AND** ADS,Y = **AND** \$aaaa,Y = 39 octet bas/octet haut
- **AND** (adresse,X) = **AND** (ADS,X) = **AND** (\$aa,X) = 21 aa
- **AND** (adresse),Y = **AND** (ADS),Y = **AND** (\$aa),Y = 31 aa

L'exécution de **AND** exerce une action sur le registre d'état : le bit Z (indicateur de résultat nul) vaudra 1 si l'accumulateur contient 0 après **AND** et 0 dans le cas contraire, tandis que le bit N (indicateur de signe) sera à 1 ou 0 selon que le bit 7 de l'accumulateur sera lui aussi à 1 ou 0 après **AND**.

Le ET logique "virtuel"

L'instruction **BIT** réalise le même traitement que **AND** mais ne remet pas le résultat dans l'accumulateur qui reste donc inchangé (d'où le qualificatif "virtuel"). En revanche, le

registre d'état se trouve affecté par BIT. D'une part, l'indicateur Z est positionné comme pour AND et, d'autre part, les bits 6 et 7 du contenu de l'adresse donnée après BIT sont copiés dans les bits 6 et 7 du registre d'état (indicateurs V et N).

Supposons ainsi que vous vouliez savoir à un instant donné si le bit 7 du contenu de l'adresse \$25 est à 0 ou 1 (nombre positif ou négatif en complément à 2), mais sans pouvoir modifier le contenu des registres A, X et Y. Après une instruction BIT \$25, un test par BPL (branchement si indicateur N = 0) ou par BMI (branchement si N = 1) vous donnera l'information souhaitée et vous permettra d'orienter vos traitements en conséquence. De même, si c'est le bit 6 qui vous intéresse, BIT \$25 suivi de BVC ou BVS conduira au résultat.

L'instruction BIT ne possède que deux formats :

- BIT adresse page 0 = BIT ADS - BIT \$aa = 24 aa
- BIT adresse quelconque = BIT ADS = BIT \$aaaa = 2C octet bas/octet haut

Le OU exclusif

L'instruction EOR réalise un traitement bit par bit avec les mêmes opérandes et opérateurs que AND : accumulateur et autre chose, avec résultat de l'opération remis dans l'accumulateur. La table de vérité de EOR est :

```
0 EOR 0 = 0
0 EOR 1 = 1
1 EOR 0 = 1
1 EOR 1 = 0
```

Un bit du résultat ne sera "vrai" que si l'un des bits de départ est faux et l'autre vrai ; un EOR sur deux bits identiques (0 ou 1) donne toujours 0.

EOR #\$FF permet de calculer le complément à 1 de l'accumulateur, par exemple :

```
01101001 EOR 11111111 →
10010110
```

L'accumulateur contenait #\$69 (01101001) et, après EOR #\$FF (11111111), tous les bits à 1 se retrouvant à 0 et réciproquement, il contiendra #\$96 (10010110). En assembleur, cela s'écrirait :

```
LDA #$69
EOR #$FF
```

En revanche, EOR #\$00 laisse le contenu de l'accumulateur inchangé, comme vous n'aurez aucune peine à le vérifier.

L'exécution de EOR affecte les indicateurs N et Z dans les mêmes conditions que celle de AND. Les

formats possibles sont :

- EOR nombre = EOR #\$NN - 49 NN
- EOR adresse page 0 = EOR ADS = EOR \$aa = 45 aa
- EOR adresse quelconque = EOR ADS = EOR \$aaaa = 4D octet bas/octet haut
- EOR adresse page 0,X = EOR ADS,X = EOR \$aa,X = 55 aa
- EOR adresse quelconque,X = EOR ADS,X = EOR \$aaaa,X = 5D octet bas/octet haut
- EOR adresse quelconque,Y = EOR ADS,Y = EOR \$aaaa,Y = 59 octet bas/octet haut
- EOR (adresse,X) = EOR (ADS,X) = EOR (\$aa,X) = 41 aa
- EOR (adresse),Y = EOR (ADS),Y = EOR (\$aa),Y = 51 aa

Le OU inclusif

C'est le même, sur le principe, que celui de l'Applesoft. Pour que le résultat soit "vrai", il suffit cette fois que l'un des deux éléments de départ le soit sans que l'autre doive nécessairement être "faux". Le code de l'instruction est ORA et elle utilise la table de vérité :

```
0 ORA 0 = 0
0 ORA 1 = 1
1 ORA 0 = 1
1 ORA 1 = 1
```

Le ORA se fait entre l'accumulateur et autre chose, le résultat étant toujours remis dans l'accumulateur.

Cette instruction permet notamment de forcer des bits à 1 sans modifier les autres. Par exemple, ORA #\$10 (00010000 en binaire) force le bit 4 de l'accumulateur à 1 :

```
10001011 ORA 00010000 →
10011011
```

En assembleur, cela s'écrirait :

```
LDA #$8B
ORA #$10
et l'on aurait alors #$9B (10011011) dans l'accumulateur.
```

A noter que si le bit 7 de l'accumulateur est à 0, ORA #\$80 (10000000)

éliminées les conséquences de la valeur de l'indicateur C, ADC #\$80 donne le même résultat que SBC #\$80 (-128 = 128 = \$80). De même, lorsque le bit 7 de l'accumulateur est à 0, ORA #\$80 revient à lui retirer #\$80 (le bit 7 passe à 1 lorsqu'on ajoute le complément à 2 de #\$80 qui n'est autre que #\$80...).

Cette équivalence entre opérateurs logiques et ADC/SBC ne peut toutefois être exploitée simplement que pour des calculs sur un octet, car AND et ORA ne manipulent pas la retenue dont on a besoin dès que l'on passe à des opérations sur deux octets.

ORA affecte le registre d'état exactement comme AND et EOR (bits N et Z) ; les formats de l'instruction sont :

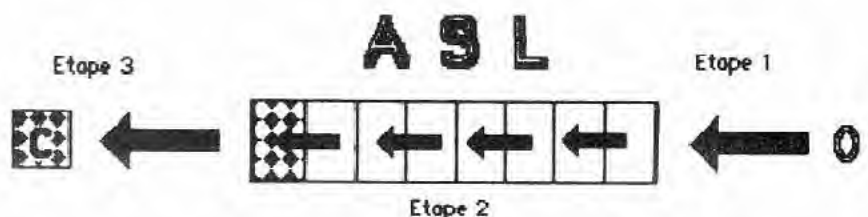
- ORA nombre = ORA #\$NN = 09 NN
- ORA adresse page 0 = ORA ADS = ORA \$aa = 05 aa
- ORA adresse quelconque = ORA ADS = ORA \$aaaa = 0D octet bas/octet haut
- ORA adresse page 0,X = ORA ADS,X = ORA \$aa,X = 15 aa
- ORA adresse quelconque,X = ORA ADS,X = ORA \$aaaa,X = 1D octet bas/octet haut
- ORA adresse quelconque,Y = ORA ADS,Y = ORA \$aaaa,Y = 19 octet bas/octet haut
- ORA (adresse,X) = ORA (ADS,X) = ORA (\$aa,X) = 01 aa
- ORA (adresse),Y = ORA (ADS),Y = ORA (\$aa),Y = 11 aa

Décalages et rotations

Décalage à gauche : ASL

L'instruction ASL agit sur un octet selon le principe suivant :

- un 0 entre dans l'octet par la droite ;
- tous les bits de l'octet sont donc poussés vers la gauche ;
- le bit 7, qui était le plus à gauche, n'a plus de place dans l'octet et il "tombe" dans le bit de retenue C.



est équivalent à CLC suivi de ADC #\$80. Si le bit 7 de l'accumulateur est à 1, vous aurez sans doute constaté que c'est AND #\$7F (01111111) qui équivaut à CLC suivi de ADC #\$80 puisque, une fois

Prenons par exemple l'octet "10100111". Si on lui fait subir un décalage à gauche, il deviendra "01001110" et le bit 7 qui était à 1 se retrouve dans la retenue : C=1 après cette opération.

Ouvrons maintenant une petite parenthèse sur le système décimal. Il est évident pour tout le monde que $2 \times 10 = 20$. Si l'on examine ce résultat de plus près, on peut dire que le chiffre 2 est en fait poussé vers la gauche par un 0 qui entre par la droite et même, en raisonnant sur un seul chiffre, qu'il tombe finalement dans un chiffre de retenue C que l'on remet ensuite devant le résultat (en binaire, une retenue ne peut valoir que 0 ou 1, mais elle peut prendre toute valeur entière entre 0 et 9 en décimal). Finalement, la multiplication par 10 en décimal n'est rien d'autre qu'un décalage des chiffres vers la gauche, poussés par un 0 qui rentre par la droite.

En binaire, et en toute logique, un décalage à gauche revient à multiplier par 2 l'octet qui fait l'objet du décalage.

Soit par exemple l'octet "01110101" (117 en décimal); à l'issue du décalage, on a "11101010" et C=0; on vérifie bien que "11101010" = 234 en décimal = 2×117 .

De même, si on décale à gauche "10100111" (167 en décimal), on obtient "01001110" (78) et C=1. Si l'on considère cette retenue de 1 comme le poids fort du résultat, on vérifie que $256 + 78 = 334 = 2 \times 167$.

Voyons maintenant les formats de l'instruction ASL avant d'examiner plus en détail le problème de la multiplication par 2 sur deux octets :

- ASL accumulateur = ASL = 0A
- ASL adresse page 0 = ASL ADS = ASL \$aa = 06 aa
- ASL adresse quelconque = ASL ADS - ASL \$aaaa = 0E octet bas/octet haut
- ASL adresse page 0,X = ASL ADS,X = ASL \$aa,X = 16 aa
- ASL adresse quelconque,X = ASL ADS,X = ASL \$aaaa,X = 1E octet bas/octet haut

Soit un nombre stocké sur deux octets O1 (poids faible) et O2 (poids fort). Multiplier ce nombre par 2 revient à :

- décaler O1 à gauche
- décaler O2 à gauche
- ajouter à l'octet O2 ainsi multiplié la retenue éventuelle provenant du décalage (multiplication) de O1.

Les lignes 1 à 17 du Programme 1 vous présentent une première méthode pour effectuer ce calcul, à l'aide des instructions que nous connaissons déjà. On initialise une variable R (retenue) à 0, mais on la fait passer à 1 si le test BCC échoue après ASL O1 (l'échec signifie que c'est un 1 qui est tombé dans la retenue C). Ensuite, on charge l'accumulateur avec O2, on le décale, on lui ajoute la valeur de R (0 ou 1) et on remet le résultat dans O2. Le sto-

ckage intermédiaire de la retenue nous est imposé par le fait que ASL O2 modifierait le bit C qui ne pourrait donc garder à coup sûr la valeur qu'il avait après ASL O1.

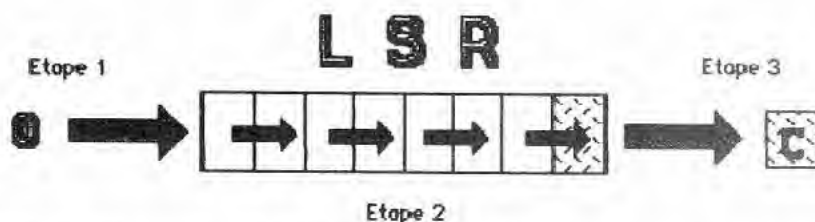
Ne regardez pas trop vite la suite du Programme 1, mais sachez néanmoins qu'il existe une manière plus simple de parvenir au même résultat...

Précisons pour finir que l'exécution de ASL agit sur le bit C, bien sûr, mais aussi sur l'indicateur N (mis à 1 si le bit 7 de l'octet est à 1 après ASL, et à 0 sinon) et l'indicateur Z (mis à 1 si l'octet ne contient plus que des 0 après ASL, à 0 sinon).

Décalage à droite : LSR

Mêmes causes, mêmes effets, mais dans l'autre sens !

- un 0 entre dans l'octet par la gauche;
- tous les bits sont repoussés vers la droite;
- le bit 0, le plus à droite, n'a plus de place et il tombe dans la retenue.



Sans revenir sur les subtilités de la division par 10, et compte tenu de ce qui précède, vous avez déjà deviné que l'instruction LSR permet notamment de diviser un octet par 2. Par exemple :

"01001110" (78 en décimal) devient "00100111" (39) après LSR, avec C=0.

"11101010" (234) devient "01110101" (117) après LSR, avec C=0.

Mais pour retrouver le résultat de $334/2$, puisque nous avions calculé tout à l'heure que $167 \times 2 = 334$, il faut de nouveau travailler sur deux octets. Nous y reviendrons après avoir donné les formats de LSR.

- LSR accumulateur = LSR = 4A
- LSR adresse page 0 = LSR ADS = LSR \$aa = 46 aa
- LSR adresse quelconque = LSR ADS = LSR \$aaaa = 4E octet bas/octet haut
- LSR adresse page 0,X = LSR ADS,X = LSR \$aa,X = 56 aa
- LSR adresse quelconque,X = LSR ADS,X = LSR \$aaaa,X = 5E octet bas/octet haut

Pour diviser par 2 sur deux octets, il faut tout d'abord diviser l'octet de

poids fort (O2) par 2. S'il est bien divisible par 2, c'est-à-dire si son bit 0 est à 0, on aura C=0 après LSR. Dans le cas contraire, on aura une retenue C=1 qu'il faudra remettre dans le bit 7 de l'octet de poids faible (O1) après avoir également divisé ce dernier par 2. Reprenons l'exemple de $334/2$ pour illustrer la chose.

Nous avons déjà vu que "10100111" (167) $\times 2 =$ "01001110" (78) plus une retenue C=1 qui est en fait un octet de poids fort "00000001".

Si l'on divise "01001110" par 2 au moyen de LSR, on obtient "00100111", soit 39 qui est bien la moitié de 78, mais, pour retrouver notre 167 de départ, il nous manque 128, soit la moitié de 256. Avec le bit 7 à 1, "00100111" devient "10100111", soit 167, et l'on peut prendre ce 1 dans C qui sera justement à 1 après le décalage à droite de "00000001".

Supposons maintenant que le poids fort soit "00000010" et le poids fai-

ble "01001110", ce qui correspond à $(2 \times 256) + 78 = 590$. La division de "00000010" par LSR donne "00000001" avec C=0, et celle de "01001110" donne toujours "00100111", d'où un résultat à ce stade de $(1 \times 256) + 39 = 295$, qui est bien la moitié de 590. Il n'est pas nécessaire cette fois de modifier le bit 7 de l'octet de poids faible, car le poids fort de départ était pair, ce que C=0 après LSR pourra nous indiquer.

Ce mécanisme des divisions par 2 serait bien sûr plus facile à analyser si l'on travaillait directement sur 16 bits. Reprenons ainsi nos deux exemples :

- $334 = 0000000101001110$
LSR 0000000101001110 donne 0000000010100111 ou encore "00000000" et "10100111"
- $590 = 0000001001001110$
LSR 0000001001001110 donne 0000000100100111 ou encore "00000001" et "00100111"

Comme nous sommes obligés de travailler sur 8 bits, c'est le bit C, après LSR sur le poids fort qui sert de "pont" entre les deux parties du nombre sur 16 bits.

Les lignes 19 à 30 du Programme 1 vous présentent une première mé-

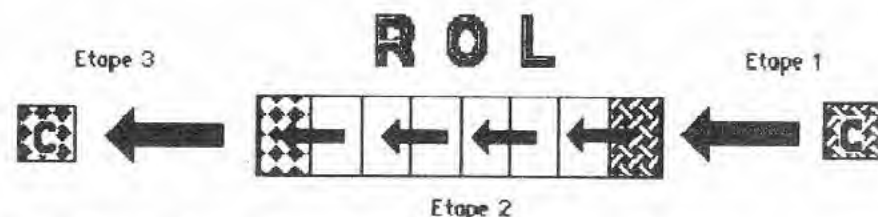
thode pour effectuer ce type de calcul. On initialise une variable R à \$80, soit 10000000, et on décale le poids fort O2 par LSR. Si le test BCS échoue, c'est que C=0 après ce premier LSR; dans ce cas, ASL R nous donnera simplement 00000000 dans R. Ensuite, on charge l'accumulateur avec le poids faible O1 et on le décale. Avec l'instruction ORA R, on fait donc ORA 10000000 si C était à 1 après LSR O2, ce qui force le bit 7 de l'accumulateur à 1, ou ORA 00000000 qui laisse l'accumulateur inchangé dans le cas contraire. Le résultat est enfin remis dans O1.

A noter que le bit N du registre d'état est toujours à 0 après LSR, puisque le bit 7 du "dernier résultat" devient justement 0. Le bit C prend la valeur du bit qui tombe dedans, et l'indicateur Z est positionné comme pour ASL.

Rotation à gauche : ROL

L'instruction ROL fonctionne comme ASL, sauf sur un point fondamental : ce n'est pas systématiquement un 0 qui rentre dans l'octet par la droite, mais la valeur du bit de retenue C (0 ou 1). Le mécanisme est donc le suivant :

- le bit de retenue C rentre par la droite;
- les bits sont poussés vers la gauche;
- le bit 7 de l'octet tombe dans la retenue.



Si le bit C est à 0 avant l'exécution de ROL, cette dernière est équivalente à ASL, mais le fait important est que ROL permet de modifier l'ordre des bits à l'intérieur d'un octet ou de faire passer des bits d'un octet à un autre.

Reprenons notre problème de la multiplication par 2 sur deux octets. Il est simple de multiplier le poids faible grâce à ASL, mais la difficulté résulte ensuite de la prise en compte d'une éventuelle retenue après la multiplication du poids fort par ce même ASL. La multiplication par 2 consiste en un décalage à gauche avec entrée d'un 0 à droite; lorsque l'on ajoute une retenue 0 ou 1 à un octet ainsi multiplié, son bit de droite ne peut prendre que les valeurs 0 ou 1 sans qu'il y ait de retenue se propageant à l'intérieur de l'octet. En d'autres termes, la multiplication du

poids fort revient à le décaler vers la gauche, puis à donner à son bit de droite la valeur de la retenue résultant du décalage du poids faible. On obtiendrait le même résultat si le bit rentrant lors du décalage était justement ce bit de retenue, et c'est exactement ce que permet de faire l'instruction ROL.

Vous trouverez cette seconde méthode de calcul dans les lignes 32 à 36 du Programme 1. Nul doute qu'elle vous paraîtra plus simple que la première.

Autre exemple d'utilisation de ROL, le découpage des octets. Supposons ainsi que vous vouliez récupérer dans un octet O2 les 4 bits de gauche d'un octet O1, qui, à eux seuls, constituent une information dont vous avez besoin (si O1=\$A2, c'est d'avoir O2=\$0A qui vous intéresse). Le petit programme suivant vous permettrait d'y parvenir :

```
LDX #4
LDA #0
STA O2
LDA O1
B0 ROL
ROL O2
DEX
BNE B0
```

Formats de l'instruction ROL :

- ROL accumulateur = ROL = 2A
- ROL adresse page 0 = ROL ADS = ROL \$aa = 26 aa
- ROL adresse quelconque = ROL

ADS = ROL \$aaaa = 2E octet bas/octet haut

- ROL adresse page 0,X = ROL ADS,X = ROL \$aa,X = 36 aa
- ROL adresse quelconque,X = ROL ADS,X = ROL \$aaaa,X = 3E octet bas/octet haut

L'exécution de ROL affecte le registre d'état comme celle de ASL.

Rotation à droite : ROR

ROR est à ROL ce que LSR est à ASL ! Cette fois, le bit C rentre par

la gauche, repousse les autres bits vers la droite et le bit 0 tombe par conséquent dans la retenue. Si C=0, ROR est équivalent à LSR.

Vous concevez sans doute, surtout si vous lisez le Programme 1 plus vite que prévu, que ROR peut nous être fort utile pour notre division par 2 sur deux octets. Tout le problème consistait justement à faire prendre au bit 7 du poids faible décalé à droite la même valeur que celle du bit C après le décalage à droite du poids fort. Il est donc bien agréable de pouvoir décaler ce poids faible en faisant rentrer à la place du bit 7 la retenue en question, comme vous le verrez dans les lignes 38 à 42 du Programme 1.

Pour en revenir maintenant au découpage des octets, si l'on veut récupérer dans O2 les 4 bits de droite de O1, il suffit de remplacer dans le programme précédent "B0 ROL" par "B0 ROR". C'est bien toujours par "ROL O2" qu'il faut faire rentrer les bits dans O2 : si vous faites "ROR O2" et si O1=\$2F, par exemple, vous retrouverez \$F0 dans O2 et non \$0F.

Formats de l'instruction ROR :

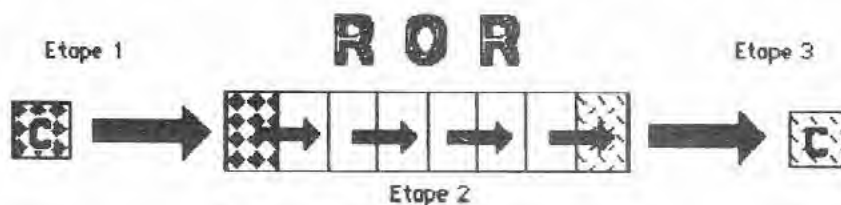
- ROR accumulateur = ROR = 6A
- ROR adresse page 0 = ROR ADS = ROR \$aa = 66 aa
- ROR adresse quelconque = ROR ADS = ROR \$aaaa = 6E octet bas/octet haut
- ROR adresse page 0,X = ROR ADS,X = ROR \$aa,X = 76 aa
- ROR adresse quelconque,X = ROR ADS,X = ROR \$aaaa,X = 7E octet bas/octet haut

L'exécution de ROR affecte les mêmes indicateurs du registre d'état que celle de ROL.

Découpage du registre d'état

Toutes ces instructions de décalage et de rotation permettent de faire sortir des bits d'un octet, de les faire rentrer dans un autre... et nous allons maintenant les utiliser pour découper et analyser un octet bien spécial : le registre d'état du micro-processeur. Pour ce faire, nous évoquerons d'abord deux nouvelles instructions tenant à l'usage de la pile.

Nous avons déjà vu qu'il était possi-



ble de déposer le contenu de l'accumulateur au sommet de la pile par PHA et de récupérer le sommet de la pile dans l'accumulateur par PLA. Ces mêmes manipulations sont possibles avec le registre d'état au moyen de PHP (08), correspondant à PHA, et de PLP (28), correspondant à PLA. Alors qu'il n'existe pas d'instruction équivalente à STA pour stocker le registre d'état dans une case-mémoire, on peut y parvenir malgré tout grâce à PHP suivi de PLA, qui permettent de récupérer le contenu du registre d'état dans l'accumulateur.

Les lignes 44 à 66 du Programme 1 constituent une sous-routine qui découpe et affiche à l'écran le registre d'état bit par bit. Nous allons en détailler le fonctionnement.

- Lignes 45 et 46 : le registre sera stocké dans la variable P dès l'appel de la routine et c'est donc le contenu du registre au moment de cet appel que nous analyserons. M sera utilisé pour l'affichage : il contiendra 0 ou 1 selon la valeur des indicateurs du registre d'état et "*" pour le bit inutilisé.

- Lignes 47 à 49 : on récupère le registre par le biais de la pile et on le stocke dans P.

- Lignes 50 à 52 : X servira de compteur de boucle (il y a 8 bits à découper) et on initialise M à 0.

- Lignes 53 et 54 : par ASL P, on fait sortir les indicateurs du registre dans l'ordre "bit 7", "bit 6", ..., "bit 0". Par ROL M, le bit sorti du registre et tombé dans la retenue rentre dans M par la droite, ce qui garantit que M contiendra toujours "00000000" ou "00000001", donc 0 ou 1.

- Lignes 55 et 56 : si X=6, on est au bit 5 du registre qui n'est pas un indicateur. On saute alors en S0 pour afficher "*" (#\$AA est le code-écran de "*" en mode NORMAL).

- Lignes 57 à 60 : pour afficher 1 ou 0, on ajoute #\$B0 au contenu de M afin d'obtenir le code-écran en mode NORMAL de "1" ou "0". On utilise ensuite la routine standard d'affichage du moniteur qui débute en \$FDF0.

- Lignes 61 à 64 : LSR M permet de faire rentrer un 0 dans M par la gauche et de faire ressortir le bit qui était entré par la droite lors du ROL M; après cette instruction, M est donc ré-initialisé à "00000000". Tant que X est différent de 0 après DEX, on boucle pour traiter tous les bits du registre en séquence. Si X=0, on sort de la routine.

Le Programme 2 sert de jeu d'essai pour la routine analysée ci-dessus. On fait exécuter certaines instructions, puis on va afficher le registre d'état qui en découle par un JSR

TEST (TEST appelle la routine du Programme 1 en \$32F, puis une routine du moniteur en \$FC62 qui envoie un RETURN à l'écran, et ce afin de n'afficher qu'un registre par ligne).

Le programme Basic P2.DEMO lance ce Programme 2 et vous donne finalement à l'écran une succession de registres dont chacun correspond à l'état des indicateurs à l'issue des instructions qui précèdent le JSR TEST ayant provoqué son affichage. Intéressez-vous surtout aux deux bits de gauche (indicateurs N et V) et aux deux bits de droite (indicateurs Z et C).

Sur la première ligne, par exemple, vous voyez :

11*10000, soit :

N=1, V=1, Z=0 et C=0.

Cet état résulte de l'opération # \$40 + # \$40, ou encore 64 + 64 en décimal, ou enfin "01000000" + "01000000" en binaire. A l'issue de cette opération, le "dernier résultat" est "10000000" : il est négatif (bit 7 à 1), d'où N=1, il y a eu débordement (retenue du bit 6 sur le bit 7), d'où V=1, le résultat est différent de 0, d'où Z=0, et il n'y a pas eu de retenue extérieure, d'où C=0.

Amusez-vous (mais oui !) à analyser de cette façon les indicateurs correspondant aux autres opérations du jeu d'essai. Ceci vous permettra de vérifier les indications "littéraires" données pour chaque instruction étudiée quant à son influence sur le registre d'état et leur donnera un caractère plus concret.

Recodage des caractères

Les codes ASCII, sous la forme desquels les caractères disponibles au clavier de votre Apple sont stockés en mémoire, sont tous inférieurs à 128. Ce n'est que pour l'affichage à l'écran et la saisie-clavier que l'on peut trouver des codes de valeur supérieure à 127.

Si l'on se rappelle que 127 = # \$7F en hexa et "01111111" en binaire, on en déduit que tous les nombres compris entre 0 et 127 en décimal ont un bit 7 à 0 et que ce bit, ayant la même valeur pour tous, ne peut servir à différencier un nombre d'un autre. En d'autres termes, 7 bits suffisent pour coder tous les caractères disponibles : le huitième n'apporte aucune information supplémentaire.

Par voie de conséquence, il doit être possible de coder 8 caractères en 7 octets : si l'on récupère pour les 7 premiers caractères la place occupée par les bits "inutiles", on dispose de 7 bits libres qui permettent de coder un huitième caractère. Cette possibilité présente un intérêt certain pour quiconque voudrait un jour dévelop-

per son propre système de gestion de fichiers de données, sans passer par DOS 3.3 ou autre, car si une disquette de 140K, par exemple, peut contenir 143360 caractères de 8 bits, elle peut stocker 163840 caractères de 7 bits. Le gain de capacité n'est pas tout à fait négligeable...

Nous nous bornerons pour l'instant à voir comment on peut recoder 8 caractères en 7 octets et les récupérer ensuite en 8 octets pour qu'ils restent exploitables par l'Applesoft, le moniteur ou le DOS.

Le programme Basic PR3.DEMO demande 8 caractères au clavier et les POKE à partir de l'adresse \$9000. Il appelle alors une routine, dont le source est Programme 3, qui recode le tout en 7 octets, les déplace à l'adresse \$9100 (afin qu'il n'y ait pas de doute sur l'efficacité du recodage) et les reconstitue en 8 octets. Le programme Basic récupère enfin les 8 caractères par des PEEKs à partir de l'adresse \$9100.

Pour vous aider dans l'analyse de la routine en assembleur, voici les principes du traitement réalisé :

- Sur le premier octet, il y a un bit à récupérer à gauche. On fait sortir le bit 7 de l'octet par ASL. Par un LSR sur le deuxième octet, on fait tomber dans la retenue son bit 0 puis, par un ROR sur le premier octet, on fait rentrer ce bit 0 du deuxième octet dans le premier tout en remettant à leur place les bits 0 à 6 de celui-ci.

- Sur le deuxième octet, il y a donc maintenant 2 bits à récupérer. Par deux ASL successifs, on fait sortir les bits 6 (ancien bit 7) et 7 (0 entré lors du LSR). En faisant ensuite deux fois "LSR sur le troisième octet suivi de ROR sur le deuxième", on récupère dans les bits 6 et 7 du deuxième les bits 0 et 1 du troisième.

- Et ainsi de suite jusqu'au septième octet sur lequel on fait sept fois ASL, puis sept fois "LSR le huitième - ROR le septième". Dans ce dernier octet, on a finalement le bit 6 du septième de départ en position 0, et les bits 0 à 6 du huitième octet aux positions 1 à 7.

- Pour retrouver 8 octets à partir de 7, on commence par faire tomber le bit 7 du premier octet dans la retenue par ASL. Puis on enchaîne ROL sur le deuxième, ROL sur le troisième... ROL sur le huitième : le bit 7 de chacun des 7 octets se retrouve ainsi en bit 0 dans le suivant. Un LSR sur le premier octet lui redonne alors sa forme de départ (remise en place des bits décalés à gauche par ASL et rentrée d'un 0 en position 7).

- Le deuxième octet a maintenant récupéré ses 7 bits "utiles" ; il lui reste à en céder un (son bit 7 actuel) qui appartient au troisième (c'était

son bit 0 de départ). On fait donc ASL sur le deuxième, suivi de ROL sur le troisième, ROL sur le quatrième... ROL sur le huitième. Un LSR sur le deuxième octet lui redonne enfin sa forme initiale.

• Ce mécanisme se poursuit jusqu'à ce que l'on ait traité de même le septième octet (ASL octet 7 - ROL octet 8 - LSR octet 7). Le huitième octet a dès lors retrouvé ses 7 bits utiles, et un AND #\$7F force son bit 7 à 0

pour garantir un code ASCII inférieur à 128. Il ne reste plus qu'à rendre la main au Basic pour qu'il récupère les 8 caractères et les affiche.

Programme 1

**END OF PASS 1
**END OF PASS 2

```
0300      1      ORG $300
0300      2      ;MULTIPLICATION PAR 2 SUR 2 OCTETS
0300      3      ;PREMIERE METHODE
0300      4      01      EPZ $6
0300      5      02      EPZ $7
0300      6      R      EPZ $8
0300      7      LDA #0
0302      8      STA R
0304      9      ASL 01
0306     10      BCC SUITE1
0308     11      INC R
030A     12      SUITE1 LDA 02
030C     13      ASL
030D     14      CLC
030E     15      ADC R
0310     16      STA 02
0312     17      RTS
0313     18      ;
0313     19      ;DIVISION PAR 2 SUR 2 OCTETS
0313     20      ;PREMIERE METHODE
0313     21      LDA #$80
0315     22      STA R
0317     23      LSR 02
0319     24      BCS SUITE2
031B     25      ASL R
031D     26      SUITE2 LDA 01
031F     27      LSR
0320     28      ORA R
0322     29      STA 01
0324     30      RTS
```

```
0325      31      ;
0325      32      ;MULTIPLICATION PAR 2 SUR 2 OCTETS
0325      33      ;DEUXIEME METHODE
0325     34      0606    34      ASL 01
0327     35      2607    35      ROL 02
0329     36      60      36      RTS
032A     37      ;
032A     38      ;DIVISION PAR 2 SUR 2 OCTETS
032A     39      ;DEUXIEME METHODE
032A     40      4607    40      LSR 02
032C     41      6606    41      ROR 01
032E     42      60      42      RTS
032F     43      ;
032F     44      ;DECOUPAGE DU REGISTRE D'ETAT
032F     45      P      EPZ $18
032F     46      M      EPZ $19
032F     47      08      47      PHP
0330     48      68      48      PLA
0331     49      8518    49      STA P
0333     50      A208    50      LDX #8
0335     51      A900    51      LDA #0
0337     52      8519    52      STA M
0339     53      0618    53      SI  ASL P
033B     54      2619    54      ROL M
033D     55      E006    55      CPX #6
033F     56      F00E    56      BEQ S0
0341     57      A519    57      LDA M
0343     58      18      58      CLC
0344     59      69B0    59      ADC #$B0
0346     60      20F0FD  60      S2  JSR $FDF0
0349     61      4619    61      LSR M
034B     62      CA      62      DEX
034C     63      D0EB    63      BNE S1
034E     64      60      64      RTS
034F     65      A9AA    65      S0  LDA #$AA
0351     66      D0F3    66      BNE S2
0353     67      ;
```

Récapitulation 1

*300.352

0300- A9 00 85 08 06 06 90 02

0308- E6 08 A5 07 0A 18 65 08
0310- 85 07 60 A9 80 85 08 46
0318- 07 B0 02 06 08 A5 06 4A
0320- 05 08 85 06 60 06 06 26
0328- 07 60 46 07 66 06 60 08

0330- 68 85 18 A2 08 A9 00 85
0338- 19 06 18 26 19 E0 06 F0
0340- 0E A5 19 18 69 B0 20 F0
0348- FD 46 19 CA D0 EB 60 A9
0350- AA D0 F3

Programme 1.DEMO

```
1  REM  DEMONSTRATION DES MULTIPLI
    CATIONS ET DIVISIONS PAR 2 SU
    R 2 OCTETS
2  REM  LA DIVISION PAR 2 D'UN NOM
    BRE IMPAIR DONNE BIEN SUR UN
    RESULTAT APPROCHE PAR DEFAUT
    (LE CALCUL NE PORTE QUE SUR D
    ES NOMBRES ENTIERS)
5  REM
6  REM
10 PRINT CHR$(4)"BLOAD PROGRAMM
    E 1.OBJ"
20 TEXT : HOME : INPUT "VOTRE NOM
    BRE : ";Z: IF Z > 32767 THEN
    20
30 POKE 7,2 / 256: POKE 6,2 - 256
    * PEEK (7): PRINT : CALL 76
    8:R = PEEK (6) + 256 * PEEK
```

```
(7): PRINT "MULTIPLICATION 1
: "R
40 POKE 7,2 / 256: POKE 6,2 - 256
    * PEEK (7): PRINT : CALL 76
    8 + 19:R = PEEK (6) + 256 *
    PEEK (7): PRINT "DIVISION 1
: "R
50 POKE 7,2 / 256: POKE 6,2 - 256
    * PEEK (7): PRINT : CALL 76
    8 + 37:R = PEEK (6) + 256 *
    PEEK (7): PRINT "MULTIPLICAT
    ION 2 : "R
60 POKE 7,2 / 256: POKE 6,2 - 256
    * PEEK (7): PRINT : CALL 76
    8 + 42:R = PEEK (6) + 256 *
    PEEK (7): PRINT "DIVISION 2
: "R
80 PRINT : PRINT "MULT. BASIC : "
    Z * 2: PRINT : PRINT "DIVISIO
    N BASIC : "Z / 2: PRINT : GET
    Z$: GOTO 20
```


LILLE..... LILLE..... LILLE..... LILLE

m.b.d.c. Apple //e, Apple ///, Mac-Intoch,
BFM 186. dragon, oric, alicé, casio

m.b.d.c. disquettes, Flexettes, 3M, Verbatim
listing, rubans encreurs.

m.b.d.c. le conseil, le matériel, les périphériques
les logiciels, les consommables,

m.b.d.c. le S.A.V. sur place.
le Service Complet.



m.b.d.c.

172, RUE SOLFERINO. 59800 LILLE — TEL. (20) 57.91.87
OUVERT DU MARDI AU SAMEDI DE 9h30 à 12h ET DE 14h30 à 19h

NAKAJIMA

AE355 : machine de bureau + imprimante ordinateur

**SEPT. 84...
LANCEMENT!**



• **AE 355, la super compact qui voit loin :**
une machine à écrire électronique de bureau
+ une imprimante pour micro-ordinateur

La NAKAJIMA AE 355 répond déjà aux exigences nouvelles de l'informatique. Elle peut être utilisée en parallèle comme imprimante terminale. De plus, elle peut être connectée en série avec un ordinateur par l'intermédiaire d'un boîtier IF-2 qui dispose d'une mémoire tampon de 2K.

• La nouvelle écriture électronique.

La machine à écrire AE 355 comporte une électronique de pointe qui en fait une machine hautement perfectionnée : introduction électronique du papier, tabulation électronique, décimale, repositionnement automatique après correction, régulation électronique de la force d'impression, centrage et souligné, marges en retrait pour paragraphes et alinéas, touche 1/2 espace... et la qualité "courrier" de ses marguerites.

• De hautes performances.

Cette machine présente une fiabilité exceptionnelle : clavier de 46 touches (100 caractères et signes), 2^e clavier pour signes et symboles particuliers, grande ligne d'écriture 292 mm, curseur de rappel des marges, mémoire de correction automatique d'une ligne complète, sélection de pas d'écriture (10, 12, 15), 6 rappels lumineux, conservation marges-tabulations : 7 jours.

KARDEX DISTRIBUTEUR NATIONAL

201, rue Carnot
94120 FONTENAY-S/BOIS
Tél. : (1) 876.11.10

BON À DÉCOUPER POUR RECEVOIR UNE DOCUMENTATION COMPLÈTE
SUR LA MACHINE À ÉCRIRE AE 355

Entreprise _____ Adresse _____ Ville _____
Nom _____ Fonction _____ Code postal _____

Programme 2

**END OF PASS 1
**END OF PASS 2

```

1000      1      ORG $1000
1000      2      ;JEU D'ESSAI POUR LE REGISTRE D'ETAT
1000      3      ;
1000 A700      4      LDA #0
1002 48        5      PHA
1003 28        6      PLP
1004 A940      7      LDA #$40
1006 18        8      CLC
1007 6940      9      ADC #$40
1009 20A410    10     JSR TEST
100C A940      11     LDA #$40
100E 38        12     SEC
100F E930      13     SBC #$30
1011 20A410    14     JSR TEST
1014 A940      15     LDA #$40
1016 38        16     SEC
1017 E950      17     SBC #$50
1019 20A410    18     JSR TEST
101C A940      19     LDA #$40
101E 38        20     SEC
101F E950      21     SBC #$50
1021 E900      22     SBC #0
1024 20A410    23     JSR TEST
1026 A900      24     LDA #0
1028 20A410    25     JSR TEST
102D A9FF      26     LDA #$FF
102D 20A410    27     JSR TEST
1030 A908      28     LDA #8
1032 C908      29     CMP #8
1034 20A410    30     JSR TEST
1037 A908      31     LDA #8
1039 C910      32     CMP #$10
103B 20A410    33     JSR TEST
103E A908      34     LDA #8
1040 C902      35     CMP #2
1042 20A410    36     JSR TEST
1045 A9FF      37     LDA #$FF
1047 297F      38     AND #$7F

```

```

1049 20A410    39     JSR TEST
104C A9FF      40     LDA #$FF
104E 2980      41     AND #$80
1050 20A410    42     JSR TEST
1053 A980      43     LDA #$80
1055 2901      44     AND #1
1057 20A410    45     JSR TEST
105A A9FF      46     LDA #$FF
105C 85CE      47     STA $CE
105E 24CE      48     BIT $CE
1060 20A410    49     JSR TEST
1063 A9FF      50     LDA #$FF
1065 A200      51     LDX #0
1067 86CE      52     STX $CE
1069 24CE      53     BIT $CE
106B 20A410    54     JSR TEST
106E A911      55     LDA #$11
1070 A911      56     EOR #$11
1072 20A410    57     JSR TEST
1075 A9FF      58     LDA #$FF
1077 4980      59     EOR #$80
1079 20A410    60     JSR TEST
107C A97F      61     LDA #$7F
107E 0980      62     ORA #$80
1080 20A410    63     JSR TEST
1083 A940      64     LDA #$40
1085 0940      65     ORA #$40
1087 20A410    66     JSR TEST
108A A940      67     LDA #$40
108C 0900      68     ORA #0
108E 20A410    69     JSR TEST
1091 A900      70     LDA #0
1093 0900      71     ORA #0
1095 20A410    72     JSR TEST
1098 A9FF      73     LDA #$FF
109A 85CE      74     STA $CE
109C C6CE      75     DEC $CE
109E 24CE      76     BIT $CE
10A0 20A410    77     JSR TEST
10A3 60        78     RTS
10A4 202F03    79     TEST JSR $32F
10A7 2062FC    80     JSR $FC62
10AA A900      81     LDA #0
10AC 48        82     PHA
10AD 28        83     PLP
10AE 60        84     RTS

```

Récapitulation 2

```

1000- A9 00 48 28 A9 40 18 69
1008- 40 20 A4 10 A9 40 38 E9
1010- 30 20 A4 10 A9 40 38 E9
1018- 50 20 A4 10 A9 40 38 E9
1020- 50 E9 00 20 A4 10 A9 00
1028- 20 A4 10 A9 FF 20 A4 10
1030- A9 08 C9 08 20 A4 10 A9
1038- 08 C9 10 20 A4 10 A9 08
1040- C9 02 20 A4 10 A9 FF 29
1048- 7F 20 A4 10 A9 FF 29 80
1050- 20 A4 10 A9 80 29 01 20
1058- A4 10 A9 FF 85 CE 24 CE
1060- 20 A4 10 A9 FF A2 00 86
1068- CE 24 CE 20 A4 10 A9 11
1070- 49 11 20 A4 10 A9 FF 49
1078- 80 20 A4 10 A9 7F 09 80
1080- 20 A4 10 A9 40 09 40 20
1088- A4 10 A9 40 09 00 20 A4
1090- 10 A9 00 09 00 20 A4 10
1098- A9 FF 85 CE C6 CE 24 CE
10A0- 20 A4 10 60 20 2F 03 20
10A8- 62 FC A9 00 48 28 60

```

Programme 2.DEMO

```

1 REM JEU D'ESSAI POUR LE DECOUP
  AGE DU REGISTRE D'ETAT
5 REM

```

```

6 REM
10 D$ = CHR$(4): PRINT D$*BLOAD
  PROGRAMME 1.OBJ": PRINT D$*BL
  OAD PROGRAMME 2.OBJ": TEXT : HOME
  : CALL 4096

```

Programme 3

**END OF PASS 1
**END OF PASS 2

```

0300      1      ORG $300
0300      2      ;
0300      3      ;CODAGE DE 8 CARACTERES EN 7 OCTETS
0300      4      SY EPZ $6
0300      5      OI EQU $9000
0300 A200      6      LDX #0
0302 A001      7      LDY #1
0304 8406      8      S3 STY SY
0306 1E0090    9      S0 ASL O1,X
0309 88        10     DEY
030A D0FA      11     BNE S0
030C A406      12     LDY SY
030E EB        13     S1 INX

```

```

030F 5E0090    14     LSR O1,X
0312 CA        15     DEX
0313 7E0090    16     ROR O1,X
0316 88        17     DEY
0317 D0F5      18     BNE S1
0319 A406      19     LDY SY
031B C8        20     INY
031C EB        21     INX
031D E007      22     CPX #7
031F D0E3      23     BNE S3
0321           24     ;
0321           25     ;TRANSFERT DES 7 OCTETS RECODES
0321           26     O EQU $9100
0321 A206      27     LDX #6
0323 8D0090    28     TO LDA O1,X
0326 9D0091    29     STA O1,X
0329 CA        30     DEX
032A 10F7      31     BPL TO
032C           32     ;

```

032C	33	;RECONSTITUTION DE 8 CARACTERES A PARTIR DE 7 OCTETS				
032C	34	SX0	EPZ	\$8		
032C	35	SX1	EPZ	\$9		
032C A200	36	LDX	#0		0343 B8	48 CLV
032E B608	37	S6	STX	SX0	0344 50F3	49 BVC S4
0330 B609	38		STX	SX1	0346 68	50 PLA
0332 E609	39		INC	SX1	0347 A608	51 LDX SX0
0334 1E0091	40		ASL	0,X	0349 5E0091	52 LSR 0,X
0337 A609	41		LDX	SX1	034C E8	53 INX
0339 3E0091	42	S4	ROL	0,X	034D E007	54 CPX #7
033C 08	43		PHP		034F 00DD	55 BNE S6
033D E8	44		INX		0351 B00091	56 LDA 0,X
033E E008	45		CPX	#8	0354 297F	57 AND #\$7F
0340 F004	46		BEQ	S5	0356 9D0091	58 STA 0,X
0342 28	47		PLP		0359 60	59 RTS

Récapitulation 3

```
0300- A2 00 A0 01 B4 06 1E 00
0308- 90 88 D0 FA A4 06 E8 5E
0310- 00 90 CA 7E 00 90 88 D0
```

```
0318- F5 A4 06 C8 E8 E0 07 D0
0320- E3 A2 06 BD 00 90 9D 00
0328- 91 CA 10 F7 A2 00 86 08
0330- 86 09 E6 09 1E 00 91 A6
0338- 09 3E 00 91 08 E8 E0 08
```

```
0340- F0 04 28 B8 50 F3 68 A6
0348- 08 5E 00 91 E8 E0 07 D0
0350- DD BD 00 91 29 7F 9D 00
0358- 91 60
```

Programme 3.DEMO

```
1LOAD PR3.DEMO
1LIST
```

```
1 REM DEMONSTRATION DU CODAGE DE
  8 CARACTERES EN 7 OCTETS
5 REM
6 REM
10 HIMEM: 9 * 4096 - 1
20 PRINT CHR$(4)"BLOAD PROGRAMM
```

```
E 3.OBJ":01 = 9 * 4096:0 = 01
+ 256
30 TEXT : HOME : FOR I = 0 TO 7: PRINT
  "CARACTERE NO "I" : "; GET C
  $(I): PRINT C$(I): POKE 01 +
  I, ASC (C$(I)): NEXT : PRINT
  : PRINT : CALL 768
40 FOR I = 0 TO 7:CC$(I) = CHR$(
  ( PEEK (0 + I)): PRINT "CARAC
  TERE NO "I" = "CC$(I): NEXT
50 PRINT : GET Z$: GOTO 30
```

Procédure &

Philippe Le Dru

Une des contraintes du Basic réside dans la définition de ses variables : en effet, toutes les variables d'un programme sont globales, c'est-à-dire que les valeurs de chacune sont, à un instant donné, identiques quelque soit le niveau où l'on se situe dans le programme.

L'utilisation de variables locales, en revanche, permet, lors de l'écriture d'un sous-programme, d'affecter les noms des variables internes à ce sous-programme sans devoir se soucier si ces noms sont déjà utilisés par ailleurs. D'où une approche modulaire plus simple des problèmes, et une programmation structurée plus efficace.

La routine de PROCEDURE s'inspire du HBASIC et de HAIFA; elle permet l'utilisation de sous-programmes

récurifs avec passage de paramètres.

Trois commandes sont disponibles :

- & HOME initialise la pile et la place sur la carte langage (pour un Apple II Plus) ou dans les 16K supplémentaires pour le //e ou le //c, pour éviter d'encombrer la mémoire centrale.
- & DEF <nom>;<variables éventuellement précédées de "!"> marque le début d'un sous-programme. Le nom doit correspondre aux règles de syntaxe des instructions DEF FN de l'Applesoft. Les variables suivant le point-virgule sont internes à l'exception de celles précédées par un point d'exclamation, qui sont alors globales.

Exemple : & DEF P1;A,B;C

Les variables globales ne peuvent

être des chaînes de caractères. Attention : celles-ci peuvent en outre être perdues si un nettoyage mémoire intervient avant leur "restauration".

- & FN <nom>;<variables ou expressions numériques> appelle le sous-programme. Les valeurs des variables ou expressions numériques sont calculées et affectées aux variables locales. L'ordre FN doit comporter le même nombre d'expressions que le sous-programme comporte de variables.

Comme dans le cas de l'instruction FN, ou comme pour les PROCEDURES en Pascal, la définition d'une procédure doit en précéder l'appel.

```
0 REM PROCEDURE.DEMO
1 REM
5 PRINT CHR$(4)"BRUNPROCEDURE.
  OBJO"
10 & HOME
11 GOSUB 100
20 FOR I = 1 TO 11
30 & FN PROC;I,TEST
```

```
34 TEST = TEST + 1
35 PRINT "GLOBAL:TEST="TEST
40 NEXT : END
100 & DEF PROC;I,TEST
120 I = I:TEST = TEST * 2
125 PRINT "PROC: TEST="TEST
130 & END
140 RETURN
```

Procédure OBJO

```

8000- A9 4C 8D F5 03 A9 10 8D
8008- F4 03 A9 80 8D F7 03 6D
8010- C9 97 D0 03 4C A6 80 C9
8018- B8 D0 03 4C 2F 80 C9 C2
8020- D0 03 4C F6 80 C9 80 D0
8028- 03 4C FD 81 4C 7C D9 20
8030- B1 00 09 80 85 14 20 EA
8038- DF 85 8A 84 8B 20 6A D0
8040- 20 D6 E3 A9 3B 20 C0 DE
8048- A5 82 48 A5 81 48 A5 B9
8050- 48 A5 88 48 A9 01 85 1C
8058- 20 B1 00 F0 08 C9 2C D0
8060- F7 E6 1C D0 F3 A5 1C 48
8068- A0 00 68 91 8A C8 C0 05
8070- D0 F8 20 B7 00 C9 00 F0
8078- 16 20 B1 00 C9 00 F0 0F
8080- C5 AF D0 F5 20 B1 00 C9
8088- 80 D0 EE 20 B1 00 60 A0
8090- 02 B1 B8 F0 0E 18 A5 B8
8098- 69 04 85 88 90 02 E6 B9
80A0- 4C 79 80 4C 10 D4 20 B1
80A8- 00 A9 00 8D DA 80 8D E5
80B0- 80 A9 E0 8D DB 80 8D E6
80B8- 80 60 CE DA 80 CE E5 80
80C0- AE DA 80 E8 D0 06 CE DB

```

```

80C8- 80 CE E6 80 AE DB 80 E0
80D0- D0 B0 06 2C 82 C0 4C 10
80D8- D4 8D 00 E0 60 AE DB 80
80E0- E0 E0 B0 EF AD 00 E0 EE
80E8- DA 80 EE E5 80 D0 06 EE
80F0- D8 80 EE E6 80 60 20 B1
80F8- 00 09 80 85 14 20 EA DF
8100- 85 8A 84 8B A0 01 B1 8A
8108- 85 1A C8 B1 8A 85 1B A9
8110- 3B 20 C0 DE A9 01 85 1C
8118- A5 88 85 18 A5 B9 85 19
8120- 20 B1 00 F0 08 C9 2C D0
8128- F7 E6 1C D0 F3 A0 00 B1
8130- 8A C5 1C F0 05 A2 2A 4C
8138- 12 D4 A5 1A 85 B8 A5 1B
8140- 85 B9 A2 00 20 B7 00 C9
8148- 21 D0 04 20 B1 00 E8 86
8150- 1D 20 E3 DF 85 85 85 06
8158- 84 66 84 07 20 B7 00 F0
8160- 03 20 BE DE A5 88 85 1A
8168- A5 B9 85 18 2C 83 C0 2C
8170- 83 C0 A0 04 B1 06 20 BA
8178- 80 98 10 F8 A5 06 20 BA
8180- 80 A5 07 20 BA 80 2C 82
8188- C0 A5 18 85 B8 A5 19 85
8190- B9 A5 1D D0 0B 85 0B 85
8198- 09 20 52 DA A5 09 F0 07

```

```

81A0- 20 E3 DF 85 08 84 09 2C
81A8- 83 C0 2C 83 C0 A5 08 20
81B0- BA 80 A5 09 20 BA 80 2C
81B8- 82 C0 20 87 00 F0 0E 20
81C0- BE DE A5 88 85 10 A5 B9
81C8- 85 19 4C 3A 81 2C 83 C0
81D0- 2C 83 C0 A5 1C 20 BA 80
81D8- A5 88 20 BA 80 A5 B9 20
81E0- BA 80 A5 75 20 BA 80 A5
81E8- 76 20 BA 80 A9 C2 20 BA
81F0- 80 2C 82 C0 A5 1A 85 B8
81F8- A5 1B 85 B9 60 20 B1 00
8200- 2C 80 C0 20 D0 90 C9 C2
8208- F0 06 2C 82 C0 4C 79 D9
8210- 20 D0 80 85 76 20 00 80
8218- 85 75 20 D0 80 85 B9 20
8220- D0 80 85 B8 20 D0 80 85
8228- 1C C9 00 D0 04 2C 82 C0
8230- 60 20 D0 80 85 09 20 D0
8238- 80 85 08 20 D0 80 85 07
8240- 20 D0 80 85 06 A5 09 F0
8248- 09 A0 05 88 B1 06 48 88
8250- 10 FA A0 00 20 D0 80 91
8258- 06 C8 C0 05 D0 F6 A5 09
8260- F0 0A A0 00 68 91 08 C8
8268- C0 05 D0 F8 C6 1C A5 1C
8270- D0 BF F0 B5

```

Programme PROCEDURE

LISA 2.5

```

1 ;*****
2 ;& HOME
3 ;& DEF NOM;A,B,1C
4 ;& END
5 ;& FN NOM ;X*Y,C(1)
6 ;*****
7      ORG $8000
8      OBJ $800
9 ;-----
10 ;VARIABLES
11 BAS      EPZ $D0
12 HAUT     EPZ $E0
13 NV       EPZ $1C
14 FN       EPZ $1A
15 TX       EPZ $18
16 WRT      EQU $C083
17 RAM      EQU $C080
18 ROM      EQU $C082
19 TP       EPZ $06
20 ;ROUTINES APPLESOFT
21 CHRGET   EQU $00B1
22 CHRGOT   EQU $00B7
23 FNGET    EQU $DFEA
24 CHKNUM   EQU $DD6A
25 ERRDIR   EQU $E3D6
26 SYNCHR   EQU $DECO
27 PTRGET   EQU $DFE3
28 CHKCOM   EQU $DEBF
29 LET      EQU $DA52
30 RETERR   EQU $D979
31 ;-----
32 ;INITIALISE &
33      LDA #$4C
34      STA $3F5
35      LDA #LABEL
36      STA $3F6
37      LDA /LABEL
38      STA $3F7
39      RTS
40 ;AIGUILLAGE
41 LABEL    CMP #$97

```

```

42      BNE >1
43      JMP ADHOME
44 *1      CMP #$B8
45      BNE >2
46      JMP ADDEF
47 *2      CMP #$C2
48      BNE >3
49      JMP ADFN
50 *3      CMP #$80
51      BNE >4
52      JMP ADEND
53 *4      JMP $D97C
54 ;-----
55 ;&DEF NOM;A,B,1C
56 ;-----
57 ADDEF     JSR CHRGET
58 ;ADRESSE  VAR.  NOM
59      ORA #$80
60      STA $14
61      JSR FNGET
62      STA $8A
63      STY $8B
64      JSR CHKNUM
65      JSR ERRDIR
66      LDA ' ;
67      JSR SYNCHR
68      LDA $82
69      PHA
70      LDA $81
71      PHA
72      LDA $B9
73      PHA
74      LDA $B8
75      PHA
76 ;NB VAR =NB DE , +1
77      LDA #1
78      STA NV
79 *1      JSR CHRGET
80      BEQ >2
81      CMP ' ,
82      BNE <1
83      INC NV
84      BNE <1
85 *2      LDA NV

```

```

86      PHA
87      LDY #0
88 *0      PLA
89      STA ($8A),Y
90      INY
91      CPY #5
92      BNE <0
93      JSR CHRGOT
94      CMP #0
95      BEQ >4
96 ;RECHERCHE &END
97 *3      JSR CHRGET
98      CMP #0
99      BEQ >4
100     CMP #$AF
101     BNE <3
102     JSR CHRGET
103     CMP #$80
104     BNE <3
105     JSR CHRGET
106     RTS
107 ;FIN LIGNE
108 *4      LDY #2
109     LDA ($B8),Y
110     BEQ >5
111     CLC
112     LDA $B8
113     ADC #4
114     STA $B8
115     BCC >6
116     INC $B9
117 *6      JMP <3
118 *5      JMP $D410
119 ;-----
120 ;&HOME
121 ;(INITIALISATION)
122 ;-----
123 ADHOME    JSR CHRGET
124      LDA #0
125      STA P1+1
126      STA P2+1
127      LDA #HAUT
128      STA P1+2
129      STA P2+2

```


130	RTS	200	;	270	;FIN D'INSTRUCTION
131	=====	201	;STOCKAGE VARIABLES	271	JSR CHRGOT
132	; SP EMPILE	202	;	272	BEQ >3
133	;	203	;RECH VAR MUETTE	273	JSR CHKCOM
134	EMPILE DEC P1+1	204	*0 LDA FN	274	;STOCKE LE PTR
135	DEC P2+1	205	STA \$B8	275	LDA \$B8
136	LDX P1+1	206	LDA FN+1	276	STA TX
137	INX	207	STA \$B9	277	LDA \$B9
138	BNE >0	208	;TEST TRANSFERT	278	STA TX+1
139	DEC P1+2	209	LDX #0	279	JMP <0
140	DEC P2+2	210	JSR CHRGOT	280	;EMPILE RETOUR
141	*0 LDX P1+2	211	CMP '!'	281	*3 BIT WRT
142	CPX #BAS	212	BNE >4	282	BIT WRT
143	BGE P1	213	JSR CHRGET	283	LDA NV
144	MEMERR BIT ROM	214	INX	284	JSR EMPILE
145	JMP \$D410	215	*4 STX \$1D	285	LDA \$B8
146	P1 STA HAUT*256	216	;POS VAR MUETTE	286	JSR EMPILE
147	RTS	217	JSR PTRGET	287	LDA \$B9
148	=====	218	STA \$85	288	JSR EMPILE
149	; SP DEPILE	219	STA \$06	289	LDA \$75
150	;	220	STY \$86	290	JSR EMPILE
151	DEPILE LDX P1+2	221	STY \$07	291	LDA \$76
152	CPX #HAUT	222	;SAUTE VIRGULE	292	JSR EMPILE
153	BGE MEMERR	223	JSR CHRGOT	293	LDA #\$C2
154	P2 LDA HAUT*256	224	BEQ >1	294	JSR EMPILE
155	INC P1+1	225	JSR CHKCOM	295	BIT ROM
156	INC P2+1	226	*1 LDA \$B8	296	;EXECUTION SP
157	BNE >0	227	STA FN	297	LDA FN
158	INC P1+2	228	LDA \$B9	298	STA \$B8
159	INC P2+2	229	STA FN+1	299	LDA FN+1
160	*0 RTS	230	;EMPILE VAR MUETTE	300	STA \$B9
161	=====	231	BIT WRT	301	RTS
162	& FN NOM ;X*Y,B(1)	232	BIT WRT	302	=====
163	;	233	LDY #4	303	; & END
164	;ADRESSE FONCTION	234	*2 LDA (\$06),Y	304	;
165	ADFN JSR CHRGET	235	JSR EMPILE	305	ADEND JSR CHRGET
166	ORA #\$80	236	DEY	306	BIT RAM
167	STA \$14	237	BPL <2	307	JSR DEPILE
168	JSR FNGET	238	;EMPILE POSITION	308	CMP #\$C2
169	STA \$8A	239	LDA \$06	309	BEQ >0
170	STY \$8B	240	JSR EMPILE	310	BIT ROM
171	LDY #1	241	LDA \$07	311	JMP RETERR
172	LDA (\$8A),Y	242	JSR EMPILE	312	*0 JSR DEPILE
173	STA FN	243	BIT ROM	313	STA \$76
174	INY	244	;RESTAURE TXTPTR	314	JSR DEPILE
175	LDA (\$8A),Y	245	LDA TX	315	STA \$75
176	STA FN+1	246	STA \$B8	316	JSR DEPILE
177	LDA ;	247	LDA TX+1	317	STA \$B9
178	JSR SYNCHR	248	STA \$B9	318	JSR DEPILE
179	;VERIF NB VARIABLES	249	;TEST	319	STA \$B8
180	LDA #1	250	LDA \$1D	320	;DEPILE NB VAR
181	STA NV	251	BNE >5	321	JSR DEPILE
182	;EMPILE TXTPTR	252	;CHARGE VAR MUETTE	322	STA NV
183	LDA \$B8	253	STA \$08	323	;SI NUL ->RETOUR
184	STA TX	254	STA \$09	324	*3 CMP #0
185	LDA \$B9	255	JSR LET	325	BNE >1
186	STA TX+1	256	LDA \$09	326	BIT ROM
187	;COMPTE LES VIRGULES	257	BEQ >6	327	RTS
188	*1 JSR CHRGET	258	;VAR A TRANSFERER	328	*1 JSR DEPILE
189	BEQ >2	259	*5 JSR PTRGET	329	STA \$09
190	CMP ;	260	STA \$08	330	JSR DEPILE
191	BNE <1	261	STY \$09	331	STA \$08
192	INC NV	262	;EMPILE POS TRANSF	332	JSR DEPILE
193	BNE <1	263	*6 BIT WRT	333	STA \$07
194	*2 LDY #0	264	BIT WRT	334	JSR DEPILE
195	LDA (\$8A),Y	265	LDA \$08	335	STA \$06
196	CMP NV	266	JSR EMPILE	336	LDA \$09
197	BEQ >0	267	LDA \$09	337	BEQ >4
198	LDX #\$2A	268	JSR EMPILE	338	;STOCKAGE PROVISIOIR
199	JMP \$D412	269	BIT ROM	339	LDY #5

RENDEZ VOTRE APPLE * ENCORE "PLUS"

Cartes et accessoires additionnels compatibles APPLE II

ENFIN UN MODEM ABORDABLE
BUZZ BOX 300 Bauds
30 cps - compatible RS 232 livré avec cor-
don et notice en français.

POUR JEUX VIDEO ET MICRO- ORDINATEURS

INTERFACE
PHS 50
UNIVERSELLE
Compatible
tous micro-ordinateurs
et jeux vidéo.
Entrée PERITEL.
Sortie UHF - SECAM L
Régulateur de tension incorporé.



449 F

FLOPPY DRIVE pour APPLE 5 POUCE

2599 F

PROMOTION DISQUETTE POUR FLOPPY

5 1/4 SF-D11 48 TPI, l'unité **21 F**
par 10 pièces l'unité **19 F** par 50 pièces l'unité **18 F**
3 1/2 double face DD 500 K octets. Unité **65 F**
5 1/4 simple face DD 80 pistes. Unité **69 F**

photo non contractuelle

**SUPER
PROMO
3 POUCE
MD3 HITACHI**
1960 F

«MONITOR BASE» SOCLE ORIENTABLE POUR MONITEURS NB ou COULEUR



S'oriente en toutes
directions •
Angle de 12,5° en
position avant et arrière
(soit 25°)
• Mobile ou fixe avec blocage.
• Patins antidérapants
• Supporte plus de 80 kg.

199 F

CARTE LANGAGE 16 K RAM



Pour extension du 48 K RAM en 64 K. Compatible
FORTRAN PASCAL, LISP, BASIC.
Entièrement équipée

649 F

CARTE D'EXTENSION 128 K RAM



Emulation disk-drive
sous UDS, PASCAL ou CP/M
Entièrement équipée

2190 F

CARTE 80 COLONNES



80 car. x 24 lignes. Résolution 7 x 9. Compatible avec
la plupart des traitements de texte BASIC,
PASCAL, CP/M, MODEM
Entièrement équipée

749 F

CARTE Z 80



Fonctionne sous CP/M
Utilisation de tout logiciel sous CP/M
Entièrement équipée

799 F

CARTE INTERFACE POUR 2 FLOPPY-DRIVE



Entièrement équipée

449 F

1^{re} Version KITS EN PROMOTION

CARTE D'UNITE CENTRALE double processeur 6502 et Z 80. 64 K RAM 7 slots d'extensions. Fonctionne sous CP/M



Entièrement équipée
(sans ROM)

3350 F

2^e Version

(modules montés, câblés, équipés)

Carte d'unité
centrale avec 6502 (sans ROM) **2290 F**

Clavier ASC II **950 F**

Alimentation 220 V. 5 A **779 F**

Coffret **698 F**



4717 F

L'ENSEMBLE **4490 F**

Chaque élément peut-être acheté séparément.

CARTE RVB



pour
moniteur
couleur

695 F

CARTE INTERFACE BUFFERISEE IMPRIMANTE



Pour toutes marques sortie CENTRONIC'S - Buffer
64 K RAM.
Livré équipée en 16 K
l'extension jusqu'à 64 K)

1690 F

CARTE INTERFACE POUR 4 IMPRIMANTES EN BATTERIE

Interface série permet de
brancher de 1 à 4 imprim-
antes

799 F

CARTE DE PROGRAMMATION 2716-2732-2764



Programmation lecture/copie
chargement de programme directement sur 2716.
Entièrement équipée

799 F

CLAVIER ASC II



66 touches. Alphanumérique.
Majuscules, minuscules, décimales.
8 touches de fonctions programmables

950 F

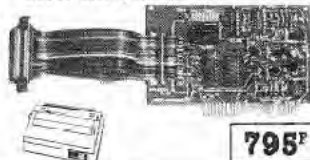
ALIMENTATION 220 V. 5 A **779 F**

COFFRET pour carte de base **698 F**

5777 F

ENSEMBLE : 5199 F

CARTE DE CONNECTION série RS 232 C



795 F

JOY-STICK



Équipé de 2 trimes
pour recherche du point zéro

**PROMO
219 F**

PROMO

169 F

TABLE GRAPHIQUE 999 F

VENTILATEUR «FAN» pour Apple



498 F

CARTE «SPEECH»



Carte langage
en Anglais et phonèmes

695 F

MONITEURS



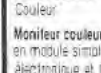
ZENITH 12"
écran vert

999 F

écran ambré

1090 F

TAXAN



Moniteur couleur RTC
en module simple à monter. Avec Peritel,
électronique et mécanique complète

3490 F

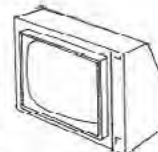
2890 F

ALBER XIV

le moniteur couleur 14"

de votre micro

Alimentation 200-240 V ± 10%
— 50/60 Hz. Puissance
consommée 40 W (nominal).
Désemination du tube image
Autonomie 200/240 V. THT 21
xV avec projection rayons X.
Signal d'entrée vidéo RVB
positif 1 V. Vidéo-carte Sensi-
ble préamplifiée. Bande pas-
sante 0,5 MHz à 5 MHz. Synchronisation niveau élevé 1,5 à
10 V, séparée du composite. Source lumineuse du néon à sélection-
née automatiquement. Niveau blanc 0,1 à 1 V compatible
avec un composant supplémentaire sans négatif. Impédance
d'entrée vidéo RVB 50 Ω. Fréquences de ligne 15 625
KHz pour 60 Hz verticalement 15 750 KHz pour 60 Hz verticalement
sélectionnée automatiquement



2990 F

EFFACEUR D'EPROM EN KIT

180 F



IMPRIMANTE GP 50A SEIKOSHA

• Entraînement à friction • Graphique
• 2 épaisseurs de caractères
• Interface parallèle
compatible CENTRONICS

1250 F

GP 500A **2390 F**
STAR GEMINI 10 x **3390 F**
EPSON H x 80 **3600 F**

ALIMENTATION A DECOUPAGE COMPATIBLE «APPLE»

Plus de puissance à 500 Watts

+ 5 V - 5 A • + 12 V - 1,5 A •
— 12 V - 0,5 A • — 5 V - 0,5 A

779 F



* APPLE est une marque déposée et appartient à APPLE COMPUTER S.A.

CONDITIONS GENERALES DE VENTES PAR CORRESPONDANCE
Pour éviter les frais de contre-remboursement, nous vous conseillons de régler vos
commandes intégralement (y compris frais de port). FORFAIT DE PORT : 20 F

ACER MICRO

42, rue de Chabrol, 75010 Paris.
Tél. 770.28.31.

340	DEY	349	INY	358	INY
341 ^5	LDA (\$06),Y	350	CPY #5	359	CPY #5
342	PHA	351	BNE <2	360	BNE <6
343	DEY	352 ;TRANSFERT		361 ;TEST FIN	
344	BPL <5	353	LDA #09	362 ^7	DEC NV
345 ;RESTAURE VAR MUET		354	BEQ >7	363	LDA NV
346 ^4	LDY #0	355	LDY #0	364	BNE <1
347 ^2	JSR DEPILE	356 ^6	PLA	365	BEQ <3
348	STA (\$06),Y	357	STA (\$08),Y	366 ;=====	
				367	END

Si l'apprentissage du Basic ne me pose pas trop de problèmes, l'utilisation de l'assembleur ne me paraît pas aussi évidente.

En conséquence, un bon exercice consiste, selon moi, à décortiquer les routines publiées dans Pom's pour les adapter à de nouvelles situations.

Ainsi, j'ai pensé qu'une modification de la routine de disquette virtuelle 16K de Michel Haag (Pom's 12) pourrait rendre quelques services aux possesseurs de carte 80 colonnes "étendue" en transformant cette réserve d'octets en un pseudo-disque bien pratique.

Les deux programmes publiés ici permettront d'atteindre cet objectif :

- La routine RWAUX proprement dite, qui sera logée à la place du secteur \$F du directory dans la carte langage, afin de la rendre indépendante des commutateurs RAMRD et RAMWRT, et qui donne accès à la pseudo-disquette.
- La routine RWAUXINIT qui se charge de la mise en place de RWAUX, du VTOC, du directory

et d'un patch du DOS qui permet l'appel du disque virtuel.

Pour plus de commodité, les deux routines sont regroupées dans un même code objet RWAUXINIT. Après l'exécution de ce programme par BRUN, tout appel au second lecteur D2 sera dirigé sur le disque virtuel logé dans l'extension mémoire de la carte 80 colonnes étendue. Le disque ainsi constitué se compose de 11 pistes complètes, d'une piste de 12 secteurs et d'une piste affectée au catalogue et à la routine d'exploitation.

Cette nouvelle version de pseudo-disquette n'est pas moins volatile que la précédente, bien sûr. Avant d'éteindre votre Apple, n'oubliez donc pas de transférer les informations stockées en RAM sur une véritable disquette dans le lecteur D1. Vous pouvez réaliser ce transfert directement à partir de vos programmes, en mode immédiat pour les programmes et fichiers binaires, ou,

de façon générale, au moyen du FID classique.

Si vous possédez deux lecteurs de disquettes, vous pouvez reconnecter le second en D2 par POKE 48372,0. Un POKE 48372,2 redirigera tout appel à D2 sur la carte d'extension.

Les pistes \$11, \$12 et \$13 se trouvant dans la carte langage, il peut être utile de verrouiller les huit derniers secteurs de la piste \$13 pour y loger une copie du moniteur, ce qui permet, par exemple, d'examiner le VTOC et le directory avec la commande "H" du moniteur étendu. Pour ce faire, "bootez" avec une disquette contenant le fichier INTBASIC puis passez en moniteur et tapez :
C08B N C08B N D084:00 00

Les secteurs \$8 à \$F de la piste \$13 sont ainsi verrouillés et le moniteur ne sera pas écrasé par les informations chargées dans le pseudo-disque.

LISA 1.5

```

1 ;*****
2 ;*
3 ;*      RWAUXINIT
4 ;*
5 ;*      J. REY (JUN 84)
6 ;*
7 ;*      (D'APRES M.HAAG)
8 ;*
9 ;*****
10 ;
11 ;
12      ORG $4F00
13 ;
14 PTR   EPZ $2A
15 ;
16 INIT  LDA $C08B      ;VALIDATION DU 1ER BANC
17      LDA $C08B
18      LDA #$00
19      STA PTR
20      LDA #$00
21      STA PTR+1
22 B1    LDA #$00
23      LDY #$00
24 B2    STA (PTR),Y

```



```

25      DEY
26      BNE B2
27      INC PTR+1
28      LDA PTR+1
29      CMP #$E0
30      BNE B1
31      ;
32      ;INITIALISATION DU VTOC
33      ;
34      LDA #$D0
35      STA PTR+1
36      LDY #$06
37 B3    LDA VTBL1,Y
38      STA (PTR),Y      ;INIT DES OCTETS $0 A $37 DU VTOC
39      DEY
40      BPL B3
41      LDA #$7A
42      STA $D027
43      LDA #$0D
44      STA $D030
45      LDA #$23
46      STA $D034
47      LDA #$10
48      STA $D035
49      LDA #$01
50      STA $D037
51      LDA #$CC
52      STA $D049      ;INHIBE LES SECTEURS 0,1,4,5 DE LA PISTE $04
53      ;
54      LDA #$FF
55      LDX #$17
56 B4    LDY VTBL2,X
57      STA (PTR),Y      ;LIBERATION DES PISTES $5,8,9,A,B,C,D,E,F,12 ET 13
58      DEX              ;PAR MISE A $FF DES OCTETS CORRESPONDANTS
59      BPL B4           ;DU VTOC
60      ;
61      LDA #$D2
62      STA PTR+1      ;INITIALISATION DES OCTETS DE
63      LDY #$01      ;CHAINAGE DES PAGES DU DIRECTORY
64 B5    LDA #$11
65      STA (PTR),Y
66      INC PTR+1
67      LDA PTR+1
68      CMP #$DF
69      BNE B5
70      DEC PTR+1
71      LDY #$02
72      LDX #$0D
73 B6    TXA
74      STA (PTR),Y
75      DEC PTR+1
76      DEX
77      BNE B6
78      ;
79      ;CHARGEMENT DE LA ROUTINE RWAX A LA PLACE DU SECTEUR $F DU DIR
80      ;
81      LDA #$DF
82      STA PTR+1
83      LDY #$00
84 B7    LDA $5000,Y
85      STA (PTR),Y
86      DEY
87      BNE B7
88      ;
89      ;DECONNECTION DE LA CARTE LANGAGE
90      ;
91      CLC
92      LDA $C082
93      LDA $C08A
94      ;
95      ;MISE EN PLACE DU PATCH PERMETTANT L'APPEL DE RWAX
96      ;

```

```

97      LDY #18
98 B8   LDA PATB,Y
99      STA $BCEB,Y
100     DEY
101     BPL B8
102     RTS
103 ;
104 ;TABLES
105 ;
106 VTBL1 HEX 04110E030000FE
107 VTBL2 HEX 31484C4D58595C5D6061646568696C6D7071747580818485EA
108 PATB  HEX 84488549A002B148C902D00DAD8BC0AD8BC04C6EDF4CEBBCEA
109     END

```

AUTO RWAUXINIT.OBJ

```

1 ;*****
2 ;*
3 ;*      RWAUX
4 ;*
5 ;*      JACQUES REY (JUN 84)
6 ;*
7 ;*      ADAPTATION DE L'IDEE DE
8 ;*      M. HAAG
9 ;*
10 ;*****
11 ;
12      ORG $DF67
13      OBJ $5067
14 ;
15 ;EMULATEUR DE DISQUETTE SUR CARTE AUXILIAIRE
16 ;
17 IOBP  EPZ $40
18 PTR   EPZ $2A
19 BUFPTR EPZ $3E
20 SECT  EPZ $2D
21 TRACK EPZ $2E
22 ;
23 TRK   EQU $4
24 SEC   EQU $5
25 BUF   EQU $8
26 CMD   EQU $0C
27 DRV   EQU $02
28 ;
29 TB1   HEX 100706      ;PISTES INTERDITES
30 TB2   HEX 00010405   ;SECTEURS INTERDITS
31 ;
32 ;TRANSFERT DES POINTEURS
33 ;
34 RWAUX LDY #BUF
35      LDA (IOBP),Y
36      STA BUFPTR
37      INY
38      LDA (IOBP),Y
39      STA BUFPTR+1
40      LDY #SEC
41      LDA (IOBP),Y
42      STA SECT
43      LDY #TRK
44      LDA (IOBP),Y
45      STA TRACK
46 ;
47 ;DETERMINATION DES PISTES ET SECTEURS
48 ;
49      CMP #14          ;ERREUR SI PISTE > #13
50      BPL ERR
51      CMP #11          ;BRANCHE SUR VTOC OU DIR
52      BEQ DIR
53      CMP #104         ;ERREUR SI PISTE < $04
54      BMI ERR

```

```

4F00- AD 8B C0 AD 8B C0 A9 00
4F08- 85 2A A9 D0 85 2B A9 00
4F10- A0 00 91 2A 88 D0 FB E6
4F18- 2B A5 2B C9 E0 D0 EF A9
4F20- D0 85 2B A0 04 B9 98 4F
4F28- 91 2A 88 10 FB A9 7A 8D
4F30- 27 D0 A9 0D 8D 3D D0 A9
4F38- 23 8D 34 D0 A9 10 8D 35
4F40- D0 A9 01 8D 37 D0 A9 CC
4F48- 8D 49 D0 A9 FF A2 17 BC
4F50- 9F 4F 91 2A CA 10 FB A9
4F58- D2 85 2B A0 01 A9 11 91
4F60- 2A E6 2B A5 2B C9 DF D0
4F68- F4 C6 2B A0 02 A2 0D 8A
4F70- 91 2A C6 2B CA D0 FB A9
4F78- DF 85 2B A0 00 B9 00 50
4F80- 91 2A 88 D0 FB 18 AD 82
4F88- C0 AD 8A C0 A0 18 B9 88
4F90- 4F 99 EB BC 88 10 F7 60
4F98- 04 11 0E 03 00 00 FE 31
4FA0- 48 4C 4D 58 59 5C 5D 60
4FA8- 61 64 65 68 69 6C 6D 70
4FB0- 71 74 75 80 81 84 85 EA
4FB8- 84 48 85 49 A0 02 B1 48
4FC0- C9 02 D0 0D AD 8B C0 AD
4FC8- 8B C0 4C 6E DF 4C EB BC
4FD0- EA EA EA EA EA EA EA EA
4FD8- EA EA EA EA EA EA EA EA
4FE0- EA EA EA EA EA EA EA EA
4FE8- EA EA EA EA EA EA EA EA
4FF0- EA EA EA EA EA EA EA EA
4FF8- EA EA EA EA EA EA EA EA
5000- 00 D2 CF D5 D4 C9 CE C5
5008- A0 A0 D2 D7 C1 D5 D8 00
5010- A0 A0 A0 A0 A0 A0 A0 A0
5018- A0 A0 A0 A0 A0 A0 A0 A0
5020- 00 CD AE C8 C1 C1 C7 A0
5028- AF A0 CA AE D2 C5 D9 00
5030- A0 A0 A0 A0 A0 A0 A0 A0
5038- A0 A0 A0 A0 A0 A0 A0 A0
5040- 00 00 00 CA D5 C9 CE A0
5048- A0 B1 B9 B8 B4 00 00 00
5050- A0 A0 A0 A0 A0 A0 A0 A0
5058- A0 A0 A0 A0 A0 A0 A0 A0
5060- 00 00 00 00 00 00 00 10
5068- 07 06 00 01 04 05 A0 08
5070- B1 48 85 3E C8 B1 48 85
5078- 3F A0 05 B1 48 85 2D A0
5080- 04 B1 48 85 2E C9 14 10
5088- 4F C9 11 F0 2C C9 04 30
5090- 47 A0 02 D9 67 DF F0 40
5098- 88 10 FB A9 00 85 2A A5
50A0- 2E 38 E9 04 0A 0A 0A 0A
50A8- 65 2D 85 2B A0 03 D9 6A
50B0- DF F0 25 88 10 FB 4C C8
50B8- DF A9 00 85 2A 18 A5 2D
50C0- C9 0F F0 14 69 D0 85 2B
50C8- A0 0C B1 48 C9 00 F0 09
50D0- C9 01 F0 0E C9 02 F0 19
50D8- 38 8D 02 C0 8D 04 C0 AD
50E0- 8A C0 A0 00 8C 03 C0 B1
50E8- 2A 91 3E 88 D0 F9 18 90
50F0- E8 A0 00 8C 05 C0 B1 3E
50F8- 91 2A 88 D0 F9 18 90

```

```

55      LDY #02      ;
56 VERIF1 CMP TB1,Y  ;
57      BEQ ERR      ;ERREUR SI PISTE = $06,$07 OU $10
58      DEY          ;(PAGE GRAPHIQUE ET PORTS)
59      BPL VERIF1    ;
60      LDA #$00
61      STA PTR
62      LDA TRACK
63      SEC
64      SBC #$04      ;
65      ASL           ;
66      ASL           ;PAGE MEMOIRE= $10*(TRACK-4)
67      ASL           ;PLUS SECT
68      ASL           ;
69      ADC SECT      ;
70      STA PTR+1
71      LDY #03      ;
72 VERIF2 CMP TB2,Y  ;
73      BEQ ERR      ;ERREUR SI PISTE $04 SECTEURS $0,$1,$4,$5
74      DEY          ;(PAGES 0,1 ET TEXTE 80COL)
75      BPL VERIF2    ;
76      JMP EXQ
77 ;
78 ;DIRECTORY
79 ;
80 DIR   LDA #$00
81      STA PTR
82      CLC
83      LDA SECT      ;CALCUL DE LA PAGE DU CATALOGUE
84      CMP #$0F      ;ERREUR SI SECTEUR $0F
85      BEQ ERR      ;(EMPLACEMENT DE LA ROUTINE)
86      ADC #$D0
87      STA PTR+1
88 ;
89 ;EXECUTION DE LA COMMANDE
90 ;
91 EXQ   LDY #CMD
92      LDA (IOBP),Y
93      CMP #$00
94      BEQ RET
95      CMP #$01
96      BEQ READ
97      CMP #$02
98      BEQ WRITE
99 ;
100 ERR  SEC          ;PROVOQUE I/O ERROR
101 ;
102 RET   STA $C002     ;REMET LES COMMUTEURS EN POSITION
103      STA $C004     ;LECT/ECRIT EN MEMOIRE PRINCIPALE
104      LDA $C08A     ;COMMUTE LA MEM EN $DFE2 QUI CONTIENT RTS
105 ;
106 READ  LDY #$00
107      STY $C003     ;MEM AUX EN LECTURE
108 BR1   LDA (PTR),Y
109      STA (BUFPTR),Y
110      DEY
111      BNE BR1
112      CLC
113      BCC RET
114 ;
115 WRITE LDY #$00
116      STY $C005     ;MEM AUX EN ECRITURE
117 BW1   LDA (BUFPTR),Y
118      STA (PTR),Y
119      DEY
120      BNE BW1
121      CLC
122      BCC RET
123      DCM "BSAVERWAUX.OBJ,A$5000,L$100"
124      END

```


Premier contact

Conception du logiciel

Appleworks est un logiciel présenté comme "intégré". Il comporte :

- un traitement de texte, parent d'Applewriter ;
- un gestionnaire de fichier, très proche de Quick File ;
- un tableur, compromis entre Visicalc (ou plutôt Magicalc) et Multiplan.

Il n'est en fait que partiellement intégré. En effet :

1 - Il est possible de "couper/coller" à l'intérieur d'un document, ou entre 2 documents DE MEME TYPE (2 tableaux par exemple), mais non entre documents de types différents (on ne peut effectuer de calculs par le tableur sur les données d'un fichier, sauf, et uniquement au moment d'une création de document nouveau, à transiter par un fichier DIF !).

2 - Il est également possible, mais moins direct, de passer d'un tableau ou d'un état composé à partir d'un fichier, vers le traitement de texte (on utilise alors une sous-option "clip board" de l'option d'impression).

La gestion du "clip board" n'est pas très simple. Par la pratique, nous avons cru comprendre que les extraits DE MEME NATURE s'entassaient et étaient ensuite repris en bloc, alors qu'un extrait d'un autre type vidait le clip board.

Tout a été fait pour faciliter l'apprentissage : nombreuses fonctions "help" à différents niveaux, identité des commandes dans les diverses composantes, utilisation relativement aisée de toutes les commandes de l'Apple //e (les 4 flèches de direction et les "pommes").

Le traitement de texte

Forces

Option "couper/coller" puissante et commode (mais avec deux modes d'utilisation différents selon qu'on travaille entre textes, ou bien entre fichier ou tableau et texte).

Désignation pratique des zones à supprimer, déplacer, dupliquer, "couper/coller".

Nombreuses options d'impression, très puissantes quand on en a bien acquis le mécanisme. Possibilité d'intervention au clavier en cours d'impression.

Faiblesses

Absence d'une fonction Glossaire.

Pas de notion de paragraphes (mais il existe un système de "marqueurs").

Quelques problèmes de justification (il n'existe pas de justification gauche

ou droite, et les options d'impression perturbent la justification totale).

Synthèse

S'il ne s'agissait que d'un nouveau traitement de texte, il n'y aurait pas de quoi déclencher l'enthousiasme.

Il faut donc voir celui-ci uniquement dans le cadre de l'intégration avec tableur et gestionnaire de fichier. Alors, on ne peut qu'être séduit par le fonctionnement du "couper/coller" direct (entre textes), ou indirect, par l'intermédiaire de l'option d'impression (entre état ou tableau et texte).

Le gestionnaire de fichier

Forces

Ce sont essentiellement celles de Quick File (simplicité et souplesse, travail très commode à l'écran, confection aisée d'états et d'étiquettes...).

A ces avantages, il convient d'ajouter bien entendu la fonction "couper/coller" dans un fichier ou entre deux fichiers et la fonction impression sur "clip board" à destination du traitement de texte.

Faiblesses

Comme les avantages, les inconvénients sont essentiellement ceux de Quick File (insuffisance de contrôles d'entrée, quelques difficultés de manipulation du curseur, capacité assez limitée).

Attention aux "sorties du système" dans le cas de certaines options imprimante non admises.

Synthèse

Plus encore que dans le cas du traitement de texte, on pourrait être tenté de dire que ce nouveau gestionnaire de fichiers est un de trop : Quick File suffisait.

Bien entendu, c'est encore dans l'intégration (entre fichiers, et avec le traitement de texte) qu'il faut en rechercher la justification.

On peut alors regretter qu'il n'y ait pas une communication directe possible avec le logiciel tableur pour effectuer facilement des calculs à partir d'éléments puisés dans un fichier et retourner les résultats vers le fichier.

Le tableur

Forces

Maniement commode de groupes de rangées, colonnes ou cases.

Appleworks à l'essai

Guy Lapautre

Possibilité de faire apparaître et d'imprimer le tableau des formules de calcul, de trier en ordre alphanumérique ou numérique dans une colonne, de rechercher une case par son contenu (alphabétique).

Et bien entendu, fonctions "couper/coller" et impression sur "clip board", ainsi que possibilité d'utiliser un format DIF (en entrée et en sortie).

Faiblesses

Pas de ligne de rappel des commandes disponibles (il faut passer par "help") et pas de "help" pour les fonctions de calcul.

Dans la duplication des formules, "no change" en option par défaut (chacun sait qu'on utilise beaucoup plus souvent "relative").

Pas de fonction de consolidation de plusieurs tableaux.

Synthèse

Un nouveau tableur, qui a pris beaucoup des fonctionnalités apparues progressivement dans ce domaine depuis "l'ancêtre" Visicalc. Il est toutefois fort regrettable que les auteurs n'aient pas cru devoir y adjoindre une fonction de consolidation.

Par ailleurs, mêmes observations que pour le gestionnaire de fichiers en ce qui concerne l'aspect d'intégration.

En résumé

Appleworks est probablement proche de ce qu'on peut faire de mieux en matière de logiciel intégré, compte tenu de la capacité et des performances de l'Apple //e, même si quelques détails irritent.

Il serait intéressant de le comparer de façon détaillée à Jane, autre logiciel intégré pour la même machine, fonctionnant selon le principe, cher à Apple, de la souris graphique exploitant un jeu d'icônes. Jane possède en outre une gestion multifenêtres.

Forces

Avant tout, d'honnêtes possibilités d'intégration entre trois logiciels : passage très simple entre des tâches de même type, assez simple entre tableur ou gestionnaire de fichiers et traitement de texte, un peu plus compliqué et moins complet entre gestionnaire de fichiers et tableur.

Fonctions communes (gestion de disque et de fichiers, hard copy, "help",...) pratiques et efficaces. Très bonne standardisation des commandes des diverses composantes.

S'agissant de chacun des logiciels :

Traitement de texte assez classique, avec de bonnes facilités pour désigner des zones de travail et de nombreuses options d'impression.

Gestionnaire de fichiers presque identique à Quick File, simple d'emploi, souple et efficace, avec confection d'états aisée.

Tableur proposant une bonne synthèse des réalisations dans ce domaine, avec une gestion commode par groupes de lignes, colonnes ou cases.

Faiblesses

On peut regretter (mais l'Apple //e est ce qu'il est), qu'il n'y ait pas une

option "couper/coller" généralisée entre les trois logiciels qui composent Appleworks. On peut également regretter qu'il n'y ait pas un quatrième logiciel, un éditeur graphique. Notons toutefois que le format DIF permet théoriquement d'utiliser Visicalc. Mais ce dernier fonctionne sous DOS 3.3...

Nous avons connu à plusieurs reprises des problèmes de sortie irrémédiable du logiciel, avec perte du dernier travail exécuté, lors de l'utilisation de certaines options d'impression.

Pour chacun des logiciels pris individuellement :

Traitement de texte dans lequel on aurait aimé retrouver le glossaire

d'Applewriter. La justification des textes est traitée de façon un peu sommaire.

Gestionnaire de fichiers un peu limité (comme Quick File) dans la structure des enregistrements, et sans contrôle d'entrées.

Tableur auquel manque essentiellement la fonction de consolidation entre divers tableaux. De ce côté, pas de progrès par rapport à Visicalc.

En synthèse

Une note BIEN, mais il manque relativement peu de choses pour en faire une note TRES BIEN.

Jane : la sauce Macintosh

Jean-Michel Gourévitch

Un traitement de texte, un tableur et une base de données réunis dans le même programme, sur la même disquette, avec des ponts permettant de passer d'une application à l'autre. Une des premières applications de la "souris", utilisable sur Apple II, II+, //e et //c, voici Jane, un logiciel à la présentation géniale, mais dont le ramage n'égale pas tout à fait, hélas, le plumage. Il a été écrit par deux petits génies de 21 ans dont l'un, Howard Marks fit ses études secondaires chez nous, au Lycée Janson de Sailly de Paris. Ce détail n'est pas inutile : il explique pourquoi ce programme développé aux Etats Unis (la société distributrice a été créée avec les fonds d'une société d'exploitation de casinos) est aussi parfaitement francisé.

Ce n'est pas la moindre des qualités d'un logiciel parfaitement "bluffant". D'abord un mot sur son aspect physique. Avouons-le tout de suite, cette Jane-là est diablement séduisante. Après une démonstration bien menée, il est difficile de ne pas craquer pour la belle...

Le manuel représenterait ce qu'on peut attendre de mieux d'un manuel (clair, illustré d'exemples et de dessins) s'il n'était parfaitement inutile. Jane est en effet livré en quatre disquettes. L'une est une auto-démonstration parfaitement menée et qui donne envie de prendre de suite la souris en main. La seconde (de couleur grise) est la disquette système. Le coeur du logiciel. La troisième (noire) est une disquette de données, qui contient aussi des exemples et des utilitaires. La dernière (jaune) est la disquette d'aide. A chaque fois qu'il en est besoin, le programme demande d'insérer la disquette nécessaire dans le second drive en la désignant par sa couleur (avis aux pirates). Si ce n'est pas la bonne, il insiste...

Mais, c'est une fois la disquette système dans le lecteur et l'Apple sous tension que le feu d'artifice commence. Jane utilise icônes et souris comme Macintosh : à propos de souris, un détail important : la dernière version de Jane fonctionne avec la souris d'Apple, et même, pour ceux qui en sont encore démunis -les possesseurs d'Apple II, II+ et //e apprécieront-, avec un simple joystick (il faut alors reconfigurer la disquette grise). Les icônes, conformes à ce nouveau langage dont raffolent les habitués des aéroports internationaux, sont plutôt bien réussies et diffèrent selon les applications (celles de la base de données sont les moins claires). On trouve, comme dans le Mac, une poubelle pour la destruction de fichiers. Et, comme avec le Lisa, une petite montre vous fait patienter lors des accès du logiciel à la disquette.

Le premier problème c'est, hélas, que cette montre est bien nécessaire. Les accès disquettes sont fréquents et lents. Une fois qu'on désigne un dossier à ouvrir en cliquant la souris, il faut par exemple une bonne dizaine de secondes avant d'y avoir accès. C'est le prix à payer pour voir s'afficher sur le moniteur d'un Apple // la réplique de l'écran de Mac tout en vert (ou presque). A partir de là, on déplace sa fenêtre sur l'écran, on ouvre de nouveaux dossiers ou de nouvelles applications (on peut entasser jusqu'à quatre dossiers sur le "bureau"). On peut trimballer un élément d'une application vers une autre, insérer par exemple une liste de chiffres du tableur dans le traitement de texte (grâce aux icônes "ciseaux", "caméra" et "pot de colle") plus facilement, il faut le reconnaître, que dans Appleworks. L'autre programme "intégré" vendu, quant à lui, par Apple. On l'aura compris,

tous ceux qui ne se sont jamais servis d'un tableur, d'un gestionnaire de fichiers ou d'un traitement de textes (beaucoup de nouveaux possesseurs d'Apple //c se reconnaîtront dans ce portrait) seront à la fête. Avec toutefois une nuance : la fréquence des accès disquette nécessite plus que pour tout autre logiciel un second drive.

Pour les autres, les vétérans d'Applewriter, les Multiplanistes (-planeurs ?) chevronnés et les CX-Multigestionnaires confirmés, Jane relèvera plus du jeu de société que de l'outil de travail. Résumons en vrac : un traitement de texte ne pouvant afficher (à condition d'avoir choisi les plus petits caractères écran) que 70 colonnes, "l'oubli" d'un glossaire, et un choix de codes d'impression étriqué ; un gestionnaire de fichiers dont les rubriques sont limitées à 25 caractères et le total des informations par saisie (rubriques comprises) à 250 caractères ; un tableur qui ne permet guère que de calculer un budget familial, sans pouvoir bien sûr lier des feuilles entre elles, sans numérotation des colonnes et avec un choix d'opérations "riquiqui"...

Pas besoin d'épiloguer, les créateurs de Jane le reconnaissent eux-mêmes, leur programme ne vise pas les "pros". Ceux qui passeraient outre risquent la déception, et accuseront ce programme "bluffant" de tricherie. Ils n'auront pas tout à fait raison. Car le mérite de Jane, c'est d'ouvrir ce bon vieux Apple II aux techniques "d'avant-garde" de l'intégration, des souris et des fenêtres. En fait de fenêtre, celle qu'ouvre Jane sur l'avenir est un apéritif tentateur. On se prend à rêver d'un logiciel de même principe tournant sur un Apple II muni d'un nouveau processeur 16 bits (le 65816, par exemple, sur lequel travaille Steve Wozniak : voir la rubrique Micro-informations) et d'une mémoire portée à 512 K. Il n'aurait guère à envier à Lotus 1-2-3 "le magnifique". Et si c'était pour demain ?

Amper-interpréteur ICARE

Pascal Cantot

L'Applesoft est pauvre, c'est entendu, mais si l'on en juge par toutes les greffes qui lui ont déjà été faites par le biais de l'ampersand (&), il semble bien que, finalement, on ne prête pas qu'aux riches...

ICARE est un amper-interpréteur assez complet, dont l'idée m'a été inspirée par le programme "Print Using & Friend" de CALL A.P.P.L.E., et par le fameux "Haïta" publié dans Pom's numéro 5.

J'ai ainsi voulu réunir dans un seul et même programme de nombreuses routines, dont certaines ont été empruntées à l'article de CALL A.P.P.L.E. déjà mentionné, d'autres à Pom's, ou que j'avais développées pour mes propres besoins. On dispose de cette façon de multiples instructions nouvelles en Applesoft, en même temps et avec un point d'entrée unique. Il n'est donc plus nécessaire de se préoccuper des problèmes de compatibilité entre les différentes routines, ni de jongler avec les disquettes et les fichiers pour charger en mémoire celles dont on a besoin.

Nous examinerons d'abord les différentes instructions disponibles et leur syntaxe, avant de voir comment mettre en oeuvre ICARE.

Les instructions d'ICARE

PRINT USNG >

& PRINT USNG > chaîne:exp
(exp désigne une expression numérique)

Exemple :

```
Si A = 1234.56,  
& PRINT USNG > " .00":A  
donne 1234.56  
& PRINT USNG > "000000.00":A  
donne 001234.56  
& PRINT USNG > " .00":A  
donne 1,234.56  
& PRINT USNG > "$ .00":A  
donne $1234.56  
& PRINT USNG > "Total =  
$.00": A donne: Total =  
$1,234.56
```

```
Si A = 9506789,  
& PRINT USNG > "Martin :  
", A donne: Martin: 950-67-89
```

Auteur: Craig Peterson - CALL A.P.P.L.E.

NOISE

& NOISE exp1,exp2

Produit un son dont la durée et la

hauteur sont données par exp1 et exp2, l'une et l'autre devant être comprises entre 0 et 255.

Auteur: Craig Peterson - CALL A.P.P.L.E.

WAIT

& WAIT exp

Attend "exp" dixièmes de seconde.

Auteur: Craig Peterson - CALL A.P.P.L.E.

SWAP >

& SWAP > variable1,variable2

Echange les contenus des variables "variable1" et "variable2", qui doivent être de même type. Par exemple: & SWAP > A\$,B\$

Auteur: Craig Peterson - CALL A.P.P.L.E.

GOTO calculé

& GOTO exp



Exécute un GOTO à la ligne dont le numéro est donné par le calcul de "exp". Par exemple, si A=100, & GOTO A/2 est équivalent à GOTO 50 (voir INTEGER BASIC).

GOSUB calculé

& GOSUB exp

Exécute un GOSUB selon les mêmes règles que pour le GOTO calculé (voir INTEGER BASIC).

DEL ARRAY >

& DEL ARRAY > nom de tableau

Efface en mémoire le tableau dont le nom est précisé dans l'instruction (par exemple, & DEL ARRAY > A\$ détruit le tableau A\$). Le tableau peut être de n'importe quel type, mais il doit absolument exister.

Auteur: Craig Peterson - CALL A.P.P.L.E.

BELL

& BELL exp

Produit un son de cloche dont la durée est donnée par la valeur de "exp", qui doit être comprise entre 0 et 255.

RPT PRINT

& RPT PRINT chaîne,exp

Affiche "exp" fois la chaîne (la valeur de exp doit être comprise entre 0 et 255).

PLAY

& PLAY exp1,exp2

Joue une note de hauteur "exp1" et de durée "exp2". Il est possible de donner plusieurs notes, mais il faut alors séparer les couples note/durée par des ";". Par exemple :

```
& PLAY 192,0 ; 172,0 ; 152,0 ;  
144,0 ; 128,0 ; 115,0 ; 102,0 ; 96,0  
joue l'octave 1.
```

Pour les octaves plus aiguës, il suffit de diviser par 2 à chaque octave, et pour obtenir les dièses, il faut faire la moyenne des deux notes adjacentes.

REG

& REG

Affiche le contenu des registres du 6502.

BKGND

& BKGNDexp

Met tous les octets de la page HGR courante à "exp". Par exemple, & BKGND255 blanchit la page courante (page 1 par défaut).

HINV

& HINV

Inverse la page HGR courante (page 1 par défaut).

SYS

& SYS

Passage en mode moniteur (équivalent à CALL -151).

MOVE

& MOVE exp1,exp2 TO exp3

Fait exécuter la commande MOVE du moniteur et déplace à l'adresse "exp3" la zone de mémoire comprise entre les adresses "exp1" et "exp2".

Accès direct à la disquette

& BUFFER = exp

Sélectionne la page du buffer qui sera utilisé par RWTS dans les commandes de lecture / écriture. Par

exemple, avec & BUFFER = 32, le buffer commence en $32 * 256 = 8192$. La valeur par défaut est \$96, soit le premier buffer du DOS avec MAXFILES 3.

& READ P,S

Lit le secteur S de la piste P dans le buffer. Le lecteur utilisé est le dernier lecteur sélectionné par une commande DOS.

& WRITE P,S

Recopie le contenu du buffer sur le secteur S de la piste P. Même remarque que ci-dessus en ce qui concerne le lecteur utilisé.

RESTORE

& RESTORE exp

Ajuste le pointeur de DATA au début de la ligne dont le numéro est donné par "exp". Un message d'erreur est envoyé si la ligne n'existe pas ou ne contient pas de DATA.

SUPER PRINT

ICARE réserve une place pour l'implantation de la routine de SUPER PRINT de Denis Sureau publiée dans le Pom's 9. Vous trouverez dans la deuxième partie de l'article la procédure à suivre pour joindre cette routine aux autres.

Si elle est implantée, la syntaxe de l'instruction est :

& SUPER PRINT;A\$,X,Y+Z,chaîne

PRINT AT

& PRINT chaîne AT H,V

Affiche le contenu de la chaîne à la colonne H de la ligne V (H de 1 à 40 et V de 1 à 24). Cette instruction ne modifie pas la position du curseur.

GET

& GET X

Remet le clavier à zéro, attend qu'une touche soit pressée et met son code (inférieur à 128 - bit 7 à 0) dans X. Peut être utilisé en mode direct.

TORTUE

ICARE réserve également la possibilité d'implanter la Tortue de Jacques Duma publiée dans le Pom's 6 (voir seconde partie pour le mode opératoire).

La syntaxe de cette commande est alors :

& DO [instruction]

Pour la description des instructions de la Tortue, se reporter au Pom's 6 ou au Recueil 2.

Touches de fonction

& XFUNCT

Initialise le programme de touches de

fonction qui dévie la routine d'entrée de caractères. Pour obtenir ensuite les fonctions, il faut appuyer simultanément sur "Pomme Ouverte" (Apple //e ou //c) ou sur un bouton de manette de jeu (Apple II+), et sur la touche concernée.

En cas de Reset, le programme est déconnecté. Chaque touche "pomme" est accompagnée d'un "bip". Attention, l'instruction & XFUNCT déconnecte la routine HCURSOR qui permet d'avoir un curseur visible en HGR (voir plus loin).

Les touches de fonction disponibles sont les suivantes :

- S'il s'agit d'un chiffre n, le programme va adresser le commutateur d'affichage situé à l'adresse \$C050 + n.
- CTRL-I : envoie 8 espaces.
- # : envoie N ED24G "RETURN" (conversion hexadécimal / décimal).
- % : fait afficher la valeur de LOMEM.
- ^ : fait afficher la valeur de HIMEM.
- * : fait entrer en mode moniteur.
- / : exécute TEXT:HOME "RETURN".
- = : affiche COLOR =
- C : envoie CATALOG
- G : envoie GR
- I : fait afficher INPUT
- L : fait afficher LOAD
- N : fait afficher NEXT
- R : fait afficher RUN
- T : fait afficher & DO[
- W : fait afficher & WAIT
- Z : fait afficher & CIRCLE =
-] : fait afficher & NOISE
- Del : recule le curseur en effaçant le caractère précédent.
- ! : exécute TEXT:NORMAL
- \$: fait afficher CHR\$(
- . : affiche HPLOT
- < : affiche PEEK(
- > : affiche POKE
- B : affiche BRUN
- E : affiche HOME : POKE 33,28 ; LIST
- H : envoie HGR
- K : affiche POKE -16368,0 :
- M : affiche MID\$(
- P : affiche PRINT
- S : affiche SAVE
- U : affiche & PRINT USING >
- X : affiche & XCIRCLE =
- [: affiche & PLAY

CIRCLE

& CIRCLE = X,Y,R

Trace un cercle de rayon R et de centre de coordonnées X et Y. Les points en dehors de l'écran sont ignorés. La valeur de R doit être absolument positive.

Auteur : Dominique Bernardi - Pom's 12.

XCIRCLE

& XCIRCLE = X,Y,R

Comme précédemment, mais le cercle est rempli avec la couleur courante. Le tracé de points en dehors de l'écran donne un résultat bizarre...

Hard-Copy

& HARDCOPY

Recopie l'écran texte sur imprimante. Les caractères en INVERSE et FLASH sont recodés. La routine ne fonctionne toutefois correctement que sur une Epson équipée d'une interface Epson.

PLOT AT

& PLOT exp AT X,Y

Envoie sur l'écran texte le caractère dont le code-écran est donné par "exp", et ce à la position X-Y (X de 0 à 39 et Y de 0 à 23).

HGR PRINT

& HGR PRINT

Connecte la routine de sortie de caractères HGR qui permet d'écrire en page HGR avec des caractères graphiques comme on écrit en page TEXT. Il faut utiliser une table de 96 caractères graphiques (une table de Haifa, par exemple) dont l'adresse de début doit être placée en 206-207 (ce qui permet de placer la table elle-même n'importe où en RAM).

La table étant chargée, il suffit de faire HGR ou HGR2 après & HGR PRINT pour écrire en graphique, sans perdre pour autant "le contact" avec le moniteur et l'Applesoft : les commandes sont toujours interprétées et vous pouvez, si vous le désirez, programmer en Gothique ou en Hébreu...

NORMAL PRINT

& NORMAL PRINT

Déconnecte HGR PRINT.

TYPE

& TYPE exp

Définit la façon dont les octets sont envoyés vers la page graphique. Si on désigne par T un octet de la table de caractères et par E l'octet de l'écran qui se trouve à la position où l'on va écrire T, les différentes valeurs possibles pour "exp" correspondent à :

- 0 : affiche T
- 1 : affiche T OR A E (DRAW)
- 2 : affiche T AND E
- 3 : affiche T XOR E (XDRAW)
- 4 : laisse E (pas d'affichage)

SCROLL

& SCROLL exp1,exp2

Scrolling HGR dans les limites de la fenêtre. "exp1" détermine la direc-

tion et "exp2" le code des octets qui vont combler le vide. Les valeurs possibles pour "exp1" sont :

- 1 : gauche
- 2 : droite
- 3 : bas
- 4 : haut

HOME

& HOME

Efface la page HGR courante dans les limites de la fenêtre et positionne le curseur en haut à gauche.

Définition de caractères graphiques

& DEF CHR\$(X) # exp1, exp2 ... exp8

Redéfinit le caractère dont le code ASCII est X (65 pour "A", par exemple) avec, dans l'ordre, les valeurs "exp1" ... "exp8". Il est possible de donner moins de 8 valeurs (le caractère ne sera alors pas entièrement défini) ou plus de 8 (les caractères suivants dans la table seront alors également modifiés).

Cette commande est à utiliser dans le mode HGR PRINT et les nouveaux caractères seront directement implantés dans la table courante

USER

& USER

Exécute un "JMP \$300" (CALL 768) et permet à l'utilisateur ne possédant pas d'assembleur d'ajouter des commandes à ICARE en implantant directement un code objet à partir de l'adresse \$300.

POKE

& POKE adresse,exp

Permet de POKER une adresse sur deux octets. Par exemple, & POKE 206,16384 met en 206-207 l'adresse de début d'une table de caractères graphiques située en \$4000 (16384 en décimal).

PEEK

& PEEK

Revectorise la fonction USR vers une routine permettant de lire un PEEK sur deux octets. Par exemple, "PRINT USR(206)" après initialisation par & PEEK permet de connaître l'adresse de la table de caractères graphiques courante.

Utilisation de la routine

S.H.LAM

& EXEC * "commande moniteur"

Permet de faire exécuter des instructions moniteur à partir de l'Applesoft, et par le biais de la routine S.H.LAM, plus facilement que par lecture de DATA, suivie de POKES et d'un CALL -144.

Par exemple, & EXEC * "300<8000,

8040M" provoquera un MOVE de la portion de mémoire comprise entre les adresses \$8000 et \$8040 à l'adresse \$300.

Utilisation de ICARE

Pour initialiser ICARE, il suffit de faire un BRUN du code objet, ou de charger ce dernier en mémoire et de faire "CALL 32768". Après initialisation du vecteur ampersand et de divers registres, la main vous est rendue au niveau de l'Applesoft en mode direct, ICARE se charge à partir de l'adresse \$8000 et place lui-même le HIMEM en dessous de cette valeur.



Compte tenu de ses dimensions, il n'est pas possible de lister dans la revue l'intégralité du programme source (réalisé en assembleur BIG MAC - MERLIN, version 64K).

Vous trouverez toutefois ci-après la liste de deux routines importantes dont je pense pouvoir affirmer le caractère original, HGR PRINT et la routine de touches de fonction. Sur la disquette de ce numéro de Pom's, ces deux routines se trouvent dans le fichier source ICARE II.

Le tronc-commun de l'assemblage, qui contient la définition des étiquettes générales et appelle les deux autres parties du source, est également listé. Il se nomme ICARE 3.1.S sur la disquette.

En ce qui concerne le code-objet, l'ensemble de la récapitulation vous est donnée, à l'exception des routines déjà publiées dans Pom's (SUPER PRINT de Denis Sureau et TORTUE de Jacques Duma). Comme il se divise en plusieurs blocs, quelques précisions peuvent être utiles à sa bonne exploitation, si vous souhaitez le rentrer en machine à la main :

- De \$8000 à \$8645 : code à reprendre tel quel
- Le "trou" entre \$8646 et \$86FC

est réservé à l'implantation éventuelle du code de la routine de SUPER PRINT.

- De \$86FD à \$886D : code à reprendre tel quel.
- Le trou entre \$886E et \$8C74 est réservé à l'implantation du code de la TORTUE. Une partie de ce code doit toutefois être obligatoirement entrée en machine car elle est utilisée par d'autres instructions de ICARE. Le code correspondant est à placer aux adresses \$8AD1 à \$8AE3. Ces adresses sont identiques à celles qu'occuperait la routine concernée si l'ensemble du code de la TORTUE était implanté dans ICARE.

- De \$8C75 à \$90D2 : code à reprendre tel quel.
- Le trou entre \$90D3 et \$9252 correspond à une table utilisée par ICARE; la valeur initiale des adresses de cette table n'a pas d'importance et vous pouvez donc sauter directement à la suite du code.
- De \$9253 à \$94F5 : code à reprendre tel quel.

Implantation de SUPER PRINT

Il suffit de mettre en mémoire à partir de l'adresse \$8646 les codes correspondant aux lignes 55 à 181 du programme source publié dans Pom's 9 (SUPER PRINT est relogeable).

Implantation de la TORTUE

Le programme original a été publié dans le Pom's 6 et dans le Recueil 2. Il faut vous y reporter et lui apporter les adaptations suivantes :

- Remplacer la dernière partie "Suite de l'instruction &" par le programme "Suite instructions" listé ci-après
- Supprimer les lignes 65 à 90 du source original.
- Mettre "ORG \$886E" et assembler le nouveau source ainsi constitué.

Le code objet résultant de ces adaptations doit ensuite être placé aux adresses \$886E à \$8C74.

Les fichiers ICARE sur la disquette Pom's

Fichiers Source

— ICARE 3.1.S : tronc commun (voir listing), appelle ICARE I et ICARE II.

— ICARE I : regroupe toutes les routines autres que HGR PRINT et les touches de fonction. Ce fichier contient des instructions "DS" pour réserver la place de SUPER PRINT et TORTUE lors de l'assemblage. Vous pouvez bien sûr remplacer cela par les sources de ces deux routines : lignes 55 à 181 du source original pour SUPER PRINT, et à partir de la ligne 91 pour TORTUE (avec toutefois la modification déjà signalée pour la dernière partie "Suite instructions").

Si vous désirez effectivement remettre ces deux sources, veillez cependant à redonner aux étiquettes utilisées dans les originaux les noms des

étiquettes générales définies dans ICARE 3.1 lorsque ces noms diffèrent.

ICARE II : reprend HGR PRINT et les touches de fonction.

Pour assembler ICARE avec BIG MAC version 64K, les étapes sont les suivantes :

- Load de ICARE 3.1
- Une fois dans l'éditeur/assembleur, tapez ASM et répondez "N" à la question "UPDATE SOURCE ...".
- L'assembleur chargera à partir de la disquette les fichiers TEXT ICARE I et ICARE II en fonction de ses besoins. L'assemblage complet dure 4 à 5 minutes.

Fichier Objet

ICARE 3.1 correspond à l'assemblage des trois fichiers source et au code listé dans la revue (les "trous" sont bien sûr occupés, mais ne contiennent pas de codes significatifs).

Pour rajouter SUPER PRINT et

TORTUE à ICARE, vous pouvez donc, soit remettre directement les codes correspondant dans le code objet, soit remettre les fichiers sources adaptés de ces deux routines dans le source de ICARE (voir ci-dessus) et ré-assembler le tout.

Le programme de démonstration également listé ici vous donnera une vue synthétique de l'utilisation de ICARE (il doit être lancé après initialisation du système). Pour l'exploiter dans la version réduite, vous devrez supprimer les appels à la Tortue (& DO()) et à SUPER PRINT. En particulier, vous pouvez remplacer la ligne 12 par "12 HGR2", enlever tous les autres "& DO[]" et rajouter une ligne "280 END". Le code POM'S.CHARSET correspond au contenu d'une table de caractères graphiques utilisée dans le programme de démonstration et que vous pouvez prendre comme base, par exemple, pour définir d'autres caractères par DEF CHR\$.

Programme DEMO ICARE

```

5 TEXT : HOME
8 HIMEM: 30719
9 LOMEM: 24577
10 DS = CHR$(4): PRINT DS"BLOADCHARSET
  ,A$7800": PRINT DS"BLOADPOM'S.CH
  RSET ,A$7B00"
12 & DO'I1$: & DO'I2$
15 & POKE 206,31488
17 & HGR PRINT
18 & TYPE = 0
20 FOR I = 1 TO 12: & PRINT "I C A R E
  " AT 15,I
25 & PLAYI * 10,32
27 & WAIT 2
30 & PRINT " " AT 15,I: NEXT
35 VTAB 13: HTAB 15: PRINT "I C A R E":
  PRINT : HCOLOR= 3
37 HPLLOT 92,92 TO 172,92 TO 172,108 TO
  92,108 TO 92,92
38 HPLLOT 93,93 TO 171,93 TO 171,107 TO
  93,107 TO 93,93
40 & DO'P116,100,0;B3$
45 & WAIT 10: & HINV: & NOISE0,0: & HI
  NV
46 FOR I = 3 TO 4: HCOLOR= I: FOR J = 4
  5 TO 90 STEP 5: & CIRCLE = 132,10
  0,J: & NOISE50,50: NEXT : NEXT
50 & WAIT 5: FOR I = 1 TO 5: & SCROLL4
  ,0
51 & PLAY64,64
52 NEXT
53 & WAIT 5
55 & BKGND171: & PRINT " "
  AT 14,8: & TYPE = 3: FOR I = 1 TO
  8: & PRINT "I C A R E" AT 15,8:
  & PLAY32,64: NEXT
60 & TYPE = 0
65 POKE 34,10: POKE 35,13: POKE 32,1: P
  OKE 33,38: & HOME

```

```

70 & POKE 206,30720: VTAB 12
72 A$ = " ICARE - Amper-interpreteur App
  lesoft": GOSUB 500
75 & TYPE = 3: FOR I = 1 TO 10: VTAB 12
  : HTAB 1: PRINT "#####
  #####";: & NOISE
  96,255: NEXT
77 & TYPE = 0
80 A$ = " ***** Par Pascal CANTOT
  ": GOSUB 500
82 A$ = " ***** PROGRAMME DE DEMON
  STRATION": GOSUB 500
84 & WAIT 20: & NOISE0,0: & HOME : PR
  INT " Ce court programme va vou
  s donner un aperçu des possibili
  tes d'ICARE..."
86 & WAIT 20
100 POKE 34,0: POKE 35,24: POKE 32,0: P
  OKE 33,40
105 & DO'I3$: HCOLOR= 3: VTAB 24: & PO
  KE 206,31488: HTAB 12: PRINT "CIR
  CLE et XCIRCLE"
108 FOR J = 3 TO 4: HCOLOR= J
110 X = 5: FOR I = 3 TO 57 STEP 3: & CIR
  CLE = X,96,I: X = X + I / 3: NEXT
115 & WAIT 10: NEXT
120 POKE 35,22: & WAIT 10: & HOME
125 I = 50: FOR J = 7 TO 0 STEP - 1: HC
  OLOR= J: & XCIRCLE = 140,90,I:I =
  I * .75: NEXT
127 & WAIT 10
130 & XCIRCLE = 140,90,51: HCOLOR= 3
132 & XCIRCLE = 140,90,50: HCOLOR= 0: &
  XCIRCLE = 155,90,40
135 POKE 35,24: & WAIT 10
141 & WAIT 1
142 & PLAY128,64:108,64:115,64:115,64:1
  28,0
150 & DO'I3;B3$
155 VTAB 24: HTAB 10: PRINT "TORTUE AMP
  ERSAND": & HOME : POKE 35,22

```



```

158 FOR K = 35 TO 20 STEP - 15
160 FOR I = 1 TO 30: FOR J = 1 TO 6: &
    DO*AK;T60$: NEXT : & DO*T125: NEX
    T
162 NEXT
163 & MOVE8192,16383 TO 16384
165 & WAIT 20:I = FRE (0): FOR I = 1
    TO 31: & HINV: & BELL20: NEXT
169 T = FRE (0):T = 0
170 & WAIT 10: FOR I = 1 TO 700: POKE
    - 16299,0:T = T + T * T: POKE -
    16300,0:T = T + T * T: NEXT
172 & TYPE = 0: VTAB 24: HTAB 10: PRINT
    " SCROLLINGS ";; POKE 23
    0,64: HTAB 10: PRINT " SCROLL
    INGS ";; POKE 230,32
173 & TYPE = 0
175 & HINV: FOR I = 1 TO 25: & SCROLL2,
    255: & SCROLL3,255: & PLAY16,16:
    NEXT
180 POKE - 16299,0: POKE 230,64: & WA
    IT 10
184 J = 20
185 FOR I = 1 TO 20: POKE 32,0: POKE 33
    ,J: & SCROLL1,0: POKE 32,20: POKE
    33,20: & SCROLL2,0: & PLAY16,16:
    J = J - 1: NEXT
190 POKE 32,0: POKE 33,40: POKE 34,0: P
    OKE 35,24: HOME
195 & NORMAL PRINT
197 VTAB 8
200 INVERSE : & RPT PRINT "+++++",8: VT
    AB 18: & RPT PRINT "+++++",8: NOR
    MAL : FOR I = 8 TO 17: & PLOT 43
    AT 0,I: & PLOT 43 AT 39,I: NEXT

210 & RESTORE 9000
212 FOR I = 9 TO 15: READ AS: FOR J = 1
    TO LEN (AS): IF VAL ( MID$ (AS
    ,J,1)) THEN & PLOT 32 AT J + 5,
    I
215 NEXT : NEXT
220 POKE - 16300,0: POKE - 16303,0
230 FOR I = 1 TO 3: & RESTORE 9100
232 & WAIT 5
235 FOR J = 1 TO 69: READ N,D
240 D = D * 1.5: IF D > 255 THEN D = 0
242 IF I = 2 THEN N = N * .75
244 IF I = 1 THEN N = N * 1.25
250 & PLAYN,D
255 FOR P = 1 TO 40: NEXT
260 NEXT
270 NEXT
297 & HGR PRINT : & POKE 206,30720
300 & DO*I35: HCOLOR= 3: HPLLOT 0,0 TO 2
    79,0 TO 279,184 TO 0,184 TO 0,0
305 HPLLOT 6,7 TO 279,7 TO 279,177 TO 6,
    177 TO 6,7
307 & TYPE = 1: FOR J = 1 TO 2
310 FOR I = 1 TO 40: & PRINT "+" AT I,
    1: & PRINT "+" AT I,23: NEXT : F
    OR I = 2 TO 22: & PRINT "+" AT 1
    ,I: & PRINT "+" AT 40,I: NEXT
312 & WAIT 5
315 & TYPE = 0: INVERSE : NEXT
320 NORMAL
325 I = FRE (0)
330 POKE 32,1: POKE 33,38: POKE 34,1: P
    OKE 35,22
335 & POKE 206,31488: & PRINT "I C A
    R E" AT 15,3: & POKE 206,30720

```

```

340 VTAB 4: PRINT : & SUPER PRINT :
    Ceci n'etait qu'une demonstration
    des possibilites graphiques et"
341 PRINT "sonores de ";
342 & POKE 206,31488: & SUPER PRINT :
    ICARE": & POKE 206,30720
345 & SUPER PRINT : Car il en a beau
    coup d'autres, toutes aussi inter
    essantes, en particulier :
350 & GET X: PRINT " - PRINT USNG >":
    PRINT " - GOTO/GOSUB": PRINT " -
    READ/WRITE/BUFFER": PRINT " - XFU
    NCT"
355 PRINT : & SUPER PRINT : Ces inst
    ructions etant difficiles a mettr
    e en valeur dans une 'demo'.",
360 & SUPER PRINT : "De meme, l'instruct
    ion DEF CHR$, qui est tres puiss
    ante. Helas, je n'ai plus le temps
    d'en faire une 'demo' valable...
    "
365 PRINT : & GET I
370 & HOME : & SUPER PRINT : En att
    endant, voila un bel utilitaire.
    Et n'oubliez pas: 2 AMPER-INTERPR
    ETEURS VALENT MIEUX QU'UN!..."
375 PRINT : PRINT : PRINT TAB( 15)"Pas
    cal CANTOT": PRINT TAB( 20)"23/0
    6/1984"
377 PRINT : & GET I
400 PRINT : PRINT : HTAB 12: & POKE 20
    6,31488: PRINT "VIVE POM'S !";
405 FOR I = 1 TO 20: HTAB 17: PRINT "
    ";; & PLAY64,32: HTAB 17: PRIN
    T "POM'S";: & PLAY64,32: NEXT
407 & POKE 206,30720
410 & WAIT 50
420 & NORMAL PRINT : TEXT : HOME
490 END
500 :
505 FOR I = 1 TO LEN (AS): VTAB 12: HT
    AB 37: PRINT MID$ (AS,I,1): & SC
    ROLL1,0: & PLAY11,16: NEXT
510 HTAB 36: PRINT " ";
512 VTAB 12: HTAB 37: PRINT " ";
530 X = FRE (0): & WAIT 10
540 RETURN
9000 DATA 111001110000100011110011111
9002 DATA 01001000100101001000101
9004 DATA 01001000001000101000101
9006 DATA 0100100000100010111001111
9008 DATA 01001000001111101010001
9010 DATA 01001000101000101001001
9012 DATA 111001110010001010001011111
9100 DATA 128,32,108,64,96,32,86,48,81,
    16,86,32,96,64,115,32,128,48,136,
    16,128,32,115,64,136,32,172,64
9110 DATA 128,32,108,64,96,32,86,32,81,
    32,86,32,96,64,115,32,144,48
9120 DATA 128,16,115,32,108,64,128,32,1
    36,48,152,16,136,32,128,196
9130 DATA 72,96,72,32,76,16,86,32,96,64
    ,115,32,144,48,128,16
9140 DATA 115,32,96,64,115,32,144,48,12
    8,16,115,32,108,64
9150 DATA 128,32,128,48,136,16,128,32,1
    15,64,136,32,172,64,72,96,72,48,7
    6,16
9160 DATA 86,32,96,64,115,32,144,48,128
    ,16,115,32,108,48,115,16,128,32,1
    36,32,152,32,136,32,128,196

```

```

1 *****
2 *
3 *
4 *
5 *
6 *
7 *
8 * Par Pascal CANTOT
9 * Copyright 20/06/1984*
10 *
11 * Et CALL A.P.P.L.E., POM'S *
12 *
13 *****
14
15 ORG $8000
16
17 KEY = $6
18 ERRBYTE = $6
19 RESULT = $6
20 XTORT = $7
21 YTORT = $9
22 USR = $0A
23 VALTYP = $11
24 LONGVAR = $18
25 ADRCOM = $18
26 HAUT = $18
27 VAR = $19
28 CMD = $19
29 EDSC = $19
30 EDTL = $19
31 EDT = $1A
32 DEBLIGN = $1B
33 SDSC = $1C
34 FINLIGN = $1D
35 SFLD = $1D
36 CNTR = $1F
37 WNDLFT = $20
38 WNDWDTH = $21
39 WNDTOP = $22
40 WNDBTM = $23
41 CH = $24
42 CV = $25
43 HBASE = $26
44 BASE = $28
45 DPLC = $1F
46 KSW = $38
47 AIL = $3C
48 A2L = $3E
49 A4L = $42
50 RND = $4E
51 LINNUM = $50
52 STREND = $6D
53 HIMEM = $73
54 CURLIN = $75
55 DATLIN = $7B
56 DATPTR = $7D
57 VARNAM = $81
58 VARPNT = $83
59 FORPNT = $85
60 LOWTR = $9B
61 DPTR = $A0
62 SGN = $A2
63 CHRGET = $B1
64 CHRGOT = $B7
65 TXTPTR = $B8
66 TABLE = $CE
67 HPAG = $E6
68 XBUT = $F9
69 YBUT = $FB
70 TORTCOL = $FC
71 ANGLE = $FD
72
73 NMBR = $100
74 IN = $200
75 INTDOS = $3EA
76 AMPV = $03F5
77 DOS = $9DBF
78 RWTS = $B7R5
79
80 KBD = $C000
81 KBDSTRB = $C010
82 POMME = $C061
83 SPKR = $C030
84
85 CHKSTK = $D3D6
86 ERROR = $D412
87 FNDLIN = $D61A
88 NEWSTT = $D7D2
89 GOTO = $D941
90 ADDON = $D998
91 SAVD = $DA9A
92 SAVSTR = $DAA1
93 CRD0 = $DAFB
94 STROUT = $DB3A
95 STRPRT = $DB3D
96 OUTD0 = $DB5C
97 FRNUM = $DD67
98 CHKNUM = $DD6A
99 CHKSTR = $DD6C
100 MMCH = $DD76
101 FRMEVL = $DD7B
102 CHKCOM = $DEBE
103 SYNCHR = $DEC0
104 SNTX = $DEC9
105 PTRGET = $DFE3
106 ISLETC = $E07D
107 INT2 = $E105
108 AYINT = $E10C
109 ILLQ = $E199
110 GIUAYF = $E2F2
111 FTED = $E625
112 GETBYTC = $E6F5
113 GETBYT = $E6F8
114 CONINT = $E6FB
115 GETNUM = $E746
116 COMBYTE = $E74C
117 GETADR = $E752
118 FADDH = $E7A0
119 FSUB = $E7A7
120 FSUBT = $E7AA
121 FADD = $E7BE
122 FADDT = $E7C1
123 MUL10 = $EA39
124 FMULT = $F97F
125 FMULTT = $E982
126 CONUPK = $E9E3
127 FDIV = $EA66
128 MOVFM = $EAF9
129 MOVFMF = $EB2B
130 MOVFA = $EB53
131 RND8 = $EB72
132 ABS = $EBAF
133 INT = $EC23
134 FOUT = $ED34
135 NEGOP = $EED0
136 COS = $EFEA
137 SIN = $EFF1
138 P12 = $F06B
139 HCLR = $F3F2
140 HPLT = $F457
141 HLIN = $F53A
142 HCOLOR = $F6F0
143 GETARYPT = $F7D9
144 REGDSP = $FA07
145 BASCALC = $FBC1
146 BELL = $FBD0
147 BELL2 = $FBE4
148 VTAB = $FC22
149 MONSCR = $FC70
150 RKEY = $FD0C
151 KEYIN = $FD1B
152 CROUT = $FD8E
153 COUT = $FDED
154 COUT1 = $FDF0
155 MOVE = $FE2C
156 SETINU = $FE80
157 SETNORM = $FE84
158 PRERR = $FF2D
159 REST = $FF3F
160 SAVE = $FF4A
161 MONZ = $FF69
162 SETV = $FF58
163 SHLAM = $FF70
164
165 PUT ICARE I
166 PUT ICARE II
167
168 * I C A R E

```

```

1
2 *****
3 *
4 *
5 *
6 *
7 * (C) Pascal CANTOT - 20/06/84 *
8 *
9 *****
10
11
12 HGRPRINT JSR INITHPOS
13 LDA $37
14 CMP #HICOUT
15 BEQ HPRINT
16 LDA #HICOUT
17 STA $37
18 LDA #HICOUT
19 STA $36
20 LDA #HCURSOR
21 STA $39
22 LDA #HCURSOR
23 STA $38
24 JSR $3EA
25 HPRINT LDA #ZC
26 STA HCOUT
27 STA HCURSOR
28 RTS
29
30 HCOUT BIT COUT1
31 STA CHAR ;Sauvegarde le caractere
32
33 JSR SAVE ;et tous les registres
34 JSR RMARGIN
35 LDX #0
36 STX MASK
37 LDX $32 ;INVFLG
38 BMI HCOUT0 ;NORMAL?
39 LDA #$7F ;MASK=$7F
39 STA MASK
40 HCOUT0 LDA CHAR
41 BMI HCOUT1 ;Si NORMAL
42 LDX #$7F ;Initialise
43 STX MASK ;le masque d'inversion
44 AND #$3F ;Transforme les car.
45 CMP #$20 ;INVERSE ou FLASH
46 BCS HCOUT1
47 ORA #$C0
48 HCOUT1 CMP #$A0 ;CTRL?
49 BCC HPEND ;Oui->pas d'affichage
50
51 JSR FNDADR
52
53 LDX #8 ;Affichage
54 STX CNTR
55 LDA CV ;Position vert. *8
56 ASL
57 ASL
58 ASL
59 STA LINE
60 HLUP LDA LINE
61 JSR HPOSN

```

```

62 LDA CH
63 CLC
64 ADC WNDLFT
65 TAY
66 ADR = *+1 ;Comme CHRGET, il s'auto-modifie
67 LDA #2000 ;Valeur arbitraire
68 EOR MASK
69 TYPE STA (HBASE),Y ;STA/DRA/AND/EOR/LDA
70 HLUP1 STA (HBASE),Y
71 INC LINE
72 INC ADR
73 BNE HLUP2
74 INC ADR+1
75 HLUP2 DEC CNTR
76 BNE HLUP
77
78 HPEND LDA CHAR ;Restaure le caractere
79 LDX WNDBTM
80 DFX
81 STX LINE
82 LDX CV
83 CPX LINE ;Teste le scrolling d'ecran
84 BCC HPEND0
85 CMP #80 ;Return
86 BEQ HPEND1
87 CMP #6A ;Line feed?
88 BEU HPEND1
89 LDX CH
90 CPX WNDRIGHT
91 BCC HPEND0
92 CMP #95 ;" > "
93 BEQ HPEND1
94 TAX

```

```

95 BPL HPEND1
96 CMP #A0 ;Car. affichable?
97 BCS HPEND1
98 AND #7F
99 CMP #20
100
101 HPEND0 JSR REST ;Les registres...
102 JMP COUT1 ;Et affichage TEXT
103
104 CHAR DS 1
105 LINE DS 1
106 MASK DS 1
107 WNDRIGHT DS 1
108 TYPTBL HEX ?1113151B1 ;STA/DRA/AND/EOR/LDA (adr),Y
109
110 HPEND1 LDA #0
111 STA BYTE
112 JSR SCROLL4
113 JMP HPEND0
114
115 RMARGIN LDA WNDWIDTH
116 CLC
117 ADC WNDLFT
118 STA WNDRIGHT
119 DEC WNDRIGHT
120 RTS
121
122 FNDADR SEC
123 SBC #20
124 AND #7F
125 STA ADR
126 LDA #0 ;Calcul adresse caractere
127 STA ADR+1

```

```

128 CLC
129 ROL ADR
130 ROL ADR+1
131 ROL ADR
132 ROL ADR+1
133 ROL ADR
134 ROL ADR+1
135 LDA TABLE
136 ADC ADR
137 STA ADR
138 LDA TABLE+1
139 ADC ADR+1
140 STA ADR+1
141 RTS
142
143 *****
144
145 DEFCHR JSR CHRGET
146 CMP #'('
147 BNE FNDCHER
148 JSR FRMNUM
149 JSR CONINT
150 TXA
151 JSR FNDADR
152 LDA ADR
153 STA LINNUM
154 LDA ADR+1
155 STA LINNUM+1
156 JSR CHRGET
157 CMP #' '
158 BNE FNDCHER
159 LDA #0
160 STA CNTR
161 FNDCHRL JSR CHRGET
162 JSR FRMNUM
163 JSR CONINT
164 TXA
165 LDY CNTR
166 INC CNTR
167 STA (LINNUM),Y
168 JSR CHRGET
169 CMP #','
170 BEQ FNDCHRL
171
172 RTS
173
174 FNDCHER LDX #16 ;Sntx
175 JMP ERROR
176
177 *****
178
179 SETYP JSR FRMNUM
180 JSR CONINT
181 CPX #5
182 BCS SETYPERR
183 LDA TYPTBL,X
184 STA TYPE
185 RTS
186
187 SETYPERR LDX #53
188 JMP ERROR
189
190 *****
191

```

```

192 HCURSOR BIT KEYIN
193 JSR SAVE
194 LDA CV
195 ADR
196 ASL
197 ADR
198 CLC
199 ADC #7
200 JSR HPOSN
201 LDA CH
202 CLC
203 ADC WNDLFT
204 TAY
205 LDA (HBASE),Y
206 PHA
207 PHA (BASE),Y
208 STA CNTR
209 PHA
210 LDA #FF
211 STA (HBASE),Y
212 LDA #01111111
213 EOR (HBASE),Y
214 STA (HBASE),Y
215 HCURS INC RND
216 BNE HCURS1
217 INC RND+1
218 EOR (HBASE),Y
219 STA (HBASE),Y
220 LDA (BASE),Y
221 TAX
222 LDA CNTR
223 STA (BASE),Y
224 STX CNTR
225 HCURS1 BIT $C000
226 BPL HCURS
227 PLA
228 STA (BASE),Y
229 PLA
230 STA (HBASE),Y
231 LDA $C000
232 CMP #80
233 BNE HCURS2
234 JSR HCLREOL
235 HCURS2 JSR REST
236 JMP AUTOCOM
237
238 *****
239
240 NORMPRT LDA #4C
241 STA HCOUNT
242 STA HCURSOR
243 RTS
244
245 *****
246
247 HCLREOL JSR RMARGIN
248 LDA CV
249 ASL
250 ASL
251 ASL
252 STA CNTR
253 DEC CNTR
254 LDA CH
255 CLC

```

```

256 ADC WNDLFT
257 STA LIMITE
258 DEC LIMITE
259 LDX #8
260 INC CNTR
261 LDA CNTR
262 JSR HPOSN
263 DEX
264 BMI HEOL3
265 LDY WNDRIGHT
266 LDA #0
267 HEOL2 STA (HBASE),Y
268 DEY
269 CPY LIMITE
270 BNE HEOL2
271 BEQ HEOL1
272
273 HEOL3 RTS
274
275 *****
276
277 HGRCLR LDX WNDBTM
278 STX LINE
279 LDA #0
280 STA CH
281 TXA
282 STA CV
283 HOMLUP DEC CV
284 JSR HCLREOL
285 LDA CV
286 CMP WNDTOP
287 BNE HOMLUP
288 RTS
289
290 *****
291
292 ADK1 = #6
293 ADK2 = #8
294
295 SCROLL JSR GETNUM
296 STX BYTE
297 JSR RMARGIN
298 LDA LINNUM
299 CMP #1
300 BEQ SCROLL1
301 CMP #2
302 BEQ SCROLL2
303 CMP #3
304 BNE SCR
305 JMP SCROLL3
306 SCR CMP #4
307 BNE SCRERR
308 JMP SCROLL4 ;Ili dty
309 SCRERR LDX #53
310 JMP FRROR
311
312
313 *
314
315 SCROLL1 LDA WNDBTM
316 ASL
317 ASL
318 ASL
319 STA CNTR

```



```

320 JSR RMARGIN
321 LDA WNDTOP
322 ASL
323 ASL
324 ASL
325 STA LIMITE
326 DEC CNTR
327 LDA CNTR
328 JSR HPOSN
329 LDY WNDLFT
330 INY
331 S1*LUP LDA (HBASE),Y
332 DEY
333 STA (HBASE),Y
334 INY
335 INY
336 CPY WNDRGHT
337 BCC S1*LUP
338 LDA CNTR
339 JSR HPOSN
340 LDA BYTE
341 LDY WNDRGHT
342 STA (HBASE),Y
343 LDA CNTR
344 CMP LIMITE
345 BNE SCR1
346 RTS
347
348 SCROLL2 LDA WNDBTM
349 ASL
350 ASL
351 ASL
352 STA CNTR
353 JSR RMARGIN
354 LDA WNDTOP
355 ASL
356 ASL
357 ASL
358 STA LIMITE
359 LDY WNDLFT
360 DEY
361 STY LINE
362 SCR2 DEC CNTR
363 LDA CNTR
364 JSR HPOSN
365 LDY WNDRGHT
366 DEY
367 S2*LUP LDA (HBASE),Y
368 INY
369 STA (HBASE),Y
370 DEY
371 DEY
372 CPY LINE
373 RNF S2*LIIP
374 LDA CNTR
375 JSR HPOSN
376 LDA BYTE
377 LDY WNDLFT
378 STA (HBASE),Y
379 LDA CNTR
380 CMP LIMITE
381 BNE SCR2
382 RTS
383
384
385 SCROLL3 LDA WNDBTM
386 ASL
387 ASL
388 ASL
389 STA LARR
390 SEC
391 SBC #8
392 STA LDEP
393 LDA WNDTOP
394 ASL
395 ASL
396 ASL
397 STA LIMITE
398 JSR RMARGIN
399 LDY WNDLFT
400 DEY
401 STY LINE
402
403 S3*NEWL LDA LDEP
404 SEC
405 SBC #1
406 CMP LIMITE
407 BEQ S3*CLR
408 STA LDEP
409 JSR HPOSN
410 LDA HBASE
411 STA ADR1
412 LDA HBASE+1
413 STA ADR1+1
414 LDA LARR
415 SEC
416 SBC #1
417 STA LARR
418 LDA LARR
419 JSR HPOSN
420 LDA HBASE
421 STA ADR2
422 LDA HBASE+1
423 STA ADR2+1
424 LDY WNDRGHT
425 S3*LUP LDA (ADR1),Y
426 STA (ADR2),Y
427
428 CPY LINE
429 BNE S3*LIIP
430 BEQ S3*NEWL
431
432 S3*CLR LDA LARR
433 CMP LIMITE
434 BEQ SCR3*A
435 SEC
436 SBC #1
437 STA LARR
438 JSR HPOSN
439 LDA HBASE
440 STA ADR2
441 LDA HBASE+1
442 STA ADR2+1
443 LDA BYTE
444 LDY WNDRGHT
445 S3*CLUP STA (ADR2),Y
446 DEY
447 CPY LINE
448 BNE S3*CLUP
449 BEQ S3*CLR
450
451 SCR3*A RTS
452
453
454 SCROLL4 LDA WNDTOP
455 ASL
456 ASL
457 ASL
458 STA LIMITE
459 LDY WNDLFT
460 DEY
461 STY LINE
462 ADC #7
463 STA LDEP
464 LDA WNDBTM
465 ASL
466 ASL
467 ASL
468 STA LIMITE
469 JSR RMARGIN
470
471 S4*NEWL LDA LDEP
472 CLC
473 ADC #1
474 CMP LIMITE
475 BCS S4*CLR
476 STA LDEP
477 JSR HPOSN
478 LDA HBASE
479 STA ADR1
480 LDA HBASE+1
481 STA ADR1+1
482 LDA LARR
483 CLC
484 ADC #1
485 STA LARR
486 JSR HPOSN
487 LDA HBASE
488 STA ADR2
489 LDA HBASE+1
490 STA ADR2+1
491 LDY WNDRGHT
492 S4*LUP LDA (ADR1),Y
493 STA (ADR2),Y
494 CPY WNDLFT
495 BEQ S4*NEWL
496 DEY
497 BPL S4*LUP
498
499 S4*CLR DEC LIMITE
500 S4*CL LDA LARR
501 CMP LIMITE
502 BEQ SCR3*A
503 CLC
504 ADC #1
505 STA LARR
506 JSR HPOSN
507 LDA HBASE
508 STA ADR2
509 LDA HBASE+1
510 STA ADR2+1
511 LDA BYTE
512 LDY WNDRGHT
513 S4*CLUP STA (ADR2),Y
514 CPY WNDLFT
515 BEQ S4*CL
516 DEY
517 BPL S4*CLUP
518
519
520 BYTE HEX 00
521 LDEP HEX 00
522 LARR HEX 00
523 LIMITE DFB 191
524
525 *****
526 *
527 * Cette routine remplaçant
528 * HPOSN ($F411) calcule l'adr.
529 * de début d'une ligne HGR1 et
530 * la met dans HBASE ($26).
531 *
532 * Cette routine est beaucoup
533 * plus rapide, car elle utilise
534 * une table d'adresses, créée
535 * par INITHPOS.
536 *
537 * En revanche, la table occupe
538 * beaucoup de place en mémoire...
539 * Mais il fallait un scrolling
540 * rapide...
541 *
542 *****
543
544 INITHPOS LDA #192
545 STA CNTR
546 LDA HPAG
547 PHA
548 LDA #0
549 STA HPAG
550
551 INITHP1 DEC CNTR
552 LDA CNTR
553 JSR $F411 ;HPOSN
554 LDY CNTR
555 LDA HBASE+1
556 STA ADRH,Y
557 LDA HBASE
558 STA ADRL,Y
559 LDA UNIK
560 BNE INITHP1
561
562 PLA
563 STA HPAG
564 RTS
565
566 HPOSN TAY ;A=1 ligne
567 JSR FNDPAG
568 LDA ADRL,Y
569 STA HBASE
570 LDA ADRH,Y
571 ORA HPAG
572 STA HBASE+1
573 RTS
574
575 FNDPAG LDA HPAG
576 CMP #540
577 BEQ FNDPG
578 LDA #520
579 STA HPAG
580 FNDPG RTS
581
582
583 * Tables des adr. HGR pour HPOSN
584 * La page HGR est 1 par défaut
585 * En cas de changement de page,
586 * il faudra rappeler INITHPOS
587
588
589 ADRH DS 192
590 ADRL DS 192
591
592 *****
593
594 FPUKE JSR FRMINUM
595 JSR GETADR
596 LDA LINNUM
597 STA #8
598 LDA LINNUM+1
599 STA #7
600 JSR CHKCOM
601 JSR FRMINUM
602 JSR GETADR
603 LDY #0
604 LDA LINNUM
605 STA ($6),Y
606 LDA LINNUM+1
607 STA ($A),Y
608 INY
609 STA ($A),Y
610 RTS
611
612 *****
613
614 SETPEEK LDA #54C
615 STA USR
616 LDA #PEEK
617 STA USR+1
618 LDA #PEEK
619 STA USR+2
620 RTS
621

```

```

622 PEEK JSR GETADR
623 LDY #1
624 LDA (LINNUM),Y
625 TAX
626 DEY
627 LDA (LINNUM),Y
628 TAY
629 TXA
630 JMP GLUAYF
631
632
633
634 *****
635
636 *-----*
637 * AUTOCOM: touches*
638 * de fonctions *
639 *-----*
640 *par Pascal CANTOT*
641 *-----*
642
643 * APPLE //e: Appuyer sur
644 * [open-Apple] et la touche
645
646
647 SETCOM LDA #AUTOCOM
648 STA KSW
649 LDA #AUTOCOM
650 STA KSW+1
651 JMP INTDOS
652
653 AUTOCOM JSR KEYIN
654 STA KEY
655 LDY POMME
656 BMI COMMAND
657 LDY POMME+1
658 BMI COMMAND
659 RTS
660
661 COMMAND STX CNTR
662 LDY #24
663 JSR BELL2
664 LDY #0
665 LDX CNTR
666 COMFND LDA KEY
667 CMP #"" ;Del
668 BEQ DELCOM
669 CMP #*0*
670 BCC CFNDLUP
671 CHP #*8*
672 BCS CFNDLUP
673 SEC
674 ODC #*B0
675 TAY
676 STA #C050,Y
677 JMP AUTOCOM
678 CFNDLUP LDA COMTAB,Y
679 BEQ CNOTFND

```

```

680 CMP KEY
681 BEQ CFNDLUP
682 INY
683 INY
684 INY
685 JMP CFNDLUP
686 INY
687 LDA COMTAB,Y
688 STA ADRCOM
689 INY
690 LDA COMTAB,Y
691 STA ADRCOM+1
692 LDY #0
693 COMOUT LDA (ADRCOM),Y
694 BEQ COMEND
695 CMP #*BD
696 BEQ COMCR
697 STA IN,X
698 INX
699 INY
700 JSR COUT
701 JMP COMOUT
702 COMEND LDA #*A0
703 RTS
704
705 COMCR JMP CROUT
706
707 CNOTFND LDA #*20
708 RTS
709
710 DELCOM JSR $FC10 ;(-
711 LDA #*A0 ;(-
712 JSR COUT
713 LDA #*88 ;(-
714 RTS
715
716 COMTAB ASC "P"
717 DA COMLIB
718 ASC "C"
719 DA C1
720 ASC "E"
721 DA C2
722 ASC "*"
723 DA C3
724 ASC "T"
725 DA C4
726 ASC "I"
727 DA C5
728 ASC "E"
729 DA C6
730 ASC "Q"
731 DA C7
732 ASC "J"
733 DA C0
734 ASC "I"
735 DA C9
736 ASC "U"
737 DA C10

```

```

738 ASC "H"
739 DA C11
740 ASC "G"
741 DA C12
742 ASC "I"
743 DA C13
744 ASC "L"
745 DA C14
746 ASC "S"
747 DA C15
748 ASC "B"
749 DA C16
750 ASC "="
751 DA C17
752 ASC " "
753 DA C18
754 ASC "K"
755 DA C19
756 ASC "Q"
757 DA C20
758 ASC "\ "
759 DA C21
760 ASC "M"
761 DA C22
762 ASC "*"
763 DA C23
764 ASC "<"
765 DA C24
766 ASC ">"
767 DA C25
768 ASC "K"
769 DA C26
770 ASC "N"
771 DA C27
772 ASC "Y"
773 DA C28
774 ASC "*"
775 DA C29
776 ASC "X"
777 DA C30
778 HEX 89 ;Tab
779 DA C31
780 ASC "#"
781 DA C32
782 ASC "Z"
783 DA C33
784 ASC "X"
785 DA C34
786 ASC "W"
787 DA C35
788 ASC "/"
789 DA C36
790 ASC "?"
791 DA C37
792 ASC "*"
793 DA C38
794
795 HEX 00
796

```

```

797 *****
798
799 COMLIB ASC "PRINT"
800 HEX 00
801 C1 ASC "CATALOG"
802 HEX 0000
803 C2 ASC "HOME (POKE32,26,LIST)"
804 HEX 00
805 C3 ASC "CALL-151"
806 HEX 8000
807 C4 ASC "& DO!"
808 HEX 00
809 C5 ASC "INPUT"
810 HEX 00
811 C6 ASC "&PLAY"
812 HEX 00
813 C7 ASC "ASSEM"
814 HEX 8000
815 C8 ASC "&SUPER PRINT"
816 HEX 00
817 C9 ASC "&NOICE"
818 HEX 00
819 C10 ASC "&PRINT USNG)"
820 HEX 00
821 C11 ASC "HAR"
822 HEX 8000
823 C12 ASC "GR"
824 HEX 8000
825 C13 ASC "TEXT:NORMAL"
826 HEX 8000
827 C14 ASC "LOAD"
828 HEX 00
829 C15 ASC "SAVE"
830 HEX 00
831 C16 ASC "BRUN"
832 HEX 00
833 C17 ASC "COLOR ="
834 HEX 00
835 C18 ASC "HPI NT"
836 HEX 00
837 C19 ASC "RUN"
838 HEX 00
839 C20 ASC "&GET"

```

```

840 HEX 00
841 C21 ASC "LIST"
842 HEX 00
843 C22 ASC "MID$( "
844 HEX 00
845 C23 ASC "CHR$( "
846 HEX 00
847 C24 ASC "PEEK( "
848 HEX 00
849 C25 ASC "POKE"
850 HEX 00
851 C26 ASC "POKE-16368,0:"
852 HEX 00
853 C27 ASC "NEXT"
854 HEX 00
855 C28 ASC "CALL"
856 HEX 00
857 C29 ASC "?*HIMEM = *PEEK(115)+PEEK(116)*256!"
858 HEX 8000
859 C30 ASC "?*LOMEM = *PEEK(105)+PEEK(106)*256!"
860 HEX 8000
861 C31 ASC " "
862 HEX 00
863 C32 ASC "N ED24G"
864 HEX 8000
865 C33 ASC "&CIRCLE="
866 HEX 00
867 C34 ASC "&XCIRCLE="
868 HEX 00
869 C35 ASC "& WAIT "
870 HEX 00
871 C36 ASC "TEXT:HOME"
872 HEX 8000
873 C37 ASC " * RND(1)* "
874 HEX 00
875 C38 ASC "&PROUT "
876 HEX 00
877
878 HEX 00
879
880 *****
881

```

objet ICARE 3.1

*8000,8645

```

8000- A9 7F 85 74 A9 FF 85 73
8008- A2 02 BD 25 80 9D F5 03
8010- 95 4F CA 10 F5 A0 00 B9
8018- 84 80 F0 06 20 FD FD C8
8020- D0 F5 4C BF 9D 4C 28 80
8028- 20 B7 00 F0 D3 A2 00 86
8030- 19 A0 00 BD 1B 81 F0 2A
8038- C9 FF F0 08 D1 B8 D0 17
8040- C8 E8 D0 EF A9 73 A0 80
8048- 20 3A DB A9 00 85 50 A2
8050- FF 2U C0 83 4C BF 9D E8
8058- BD 1B 81 D0 FA E8 E6 19
8060- D0 CF 20 98 D9 06 19 A6
8068- 19 BD DC 81 48 BD DB 81
8070- 48 60 8D D5 CE CB CE CF
8078- D7 CE A0 C3 CF CD CD C1
8080- CE C4 8D 00 8D 8D A0 A0
8088- A0 A0 A0 A0 A0 A0 A0 A0
8090- A0 20 20 20 09 03 01 12
8098- 05 20 20 20 20 8D 8D C1
80A0- CD D0 C5 D2 AD C9 CE D4
80A8- C5 D2 D0 D2 C5 D4 C5 D5
80B0- D2 A0 C1 D0 D0 CC C5 D3
80B8- CF C6 D4 8D 8D A8 C3 A9
80C0- A0 D0 C1 D3 C3 C1 CC A0
80C8- C3 C1 CE D4 CF D4 A0 AD
80D0- A0 B1 B9 B8 B4 A0 AD A0
80D8- C5 D4 A0 CC C5 D3 8D A0
80E0- A0 A0 A0 A0 A0 A0 A0 A0
80E8- A0 A0 A0 C3 CF AD C1 D5
80F0- D4 C5 D5 D2 D3 8D A0 A0
80F8- A0 A0 A0 A0 A0 A0 A0 A0
8100- C3 C1 CC CC A0 C1 AE D0
8108- AE D0 AE CC AE C5 AE A0
8110- C5 D4 A0 D0 CF CD A7 D3
8118- AE 8D 00 BA 55 53 4E 47
8120- CF 00 42 45 4C 4C 00 52
8128- 50 54 BA 00 53 57 41 50
8130- CF 00 85 41 52 52 41 59
8138- CF 00 4E 4F 49 53 45 00
8140- B5 00 50 4C 41 59 00 AB
8148- 00 B0 00 8D 00 50 52 4F
8150- 55 54 00 48 41 52 44 43
8158- 4F 50 59 00 42 48 47 4E
8160- 44 00 48 49 4E 56 00 52
8168- 45 47 00 53 59 53 00 4D
8170- 4F 56 45 00 42 55 46 46
8178- 45 52 D0 00 87 00 57 52
8180- 49 54 45 00 AE 00 53 55
8188- 50 45 52 BA 00 BA 00 BE
8190- 00 43 49 52 43 4C 45 D0
8198- 00 58 43 49 52 43 4C 45
81A0- D0 00 91 BA 00 53 43 52
81A8- 4F 4C 4C 00 B8 E7 00 54
81B0- 59 50 45 D0 00 97 00 9D
81B8- BA 00 B9 00 E2 00 53 48
81C0- 4F 4F 54 00 44 4F 5B 00
81C8- 45 58 45 43 CA 00 58 46
81D0- 55 4E 43 54 00 55 53 45
81D8- 52 00 FF 2A 82 FD 83 26
81E0- 83 55 83 8F 83 BC 83 08
81E8- 84 24 84 65 84 78 84 D8
81F0- 83 A0 84 DA 84 1F 85 47
81F8- 85 D6 FA 68 FF 5B 85 EE
8200- 85 CA 85 E3 85 12 86 42
8208- 86 FC 86 50 87 71 87 5F
8210- 88 CA 8C DC 8E D8 8D 1B
8218- 8E C4 8E 88 8E 52 92 75
8220- 92 48 84 67 88 74 8C 92
8228- 92 FF 02 20 7B DD 20 6C
8230- DD A9 1C A0 00 85 85 84
8238- 86 20 9A DA A9 3B 20 C0
8240- DE A9 1C A0 00 85 A0 84
8248- A1 A9 19 85 85 84 86 20
8250- A1 DA A2 00 A4 19 F0 1C
8258- 88 B1 1A C9 30 F0 10 C9
8260- 20 F0 0C C9 24 F0 08 C9

```

```

8268- 2E F0 09 C9 2A D0 01 E8
8270- 98 D0 E5 AA 86 1F 20 67
8278- DD A5 1F F0 07 20 39 EA
8280- C6 1F D0 F9 20 72 EB A5
8288- A2 48 20 AF EB 20 A0 E7
8290- 20 23 EC 68 10 03 20 D0
8298- EE 20 34 ED B8 A2 FF F8
82A0- BD 01 01 D0 FA A4 19 F0
82A8- 5A 98 F0 4E 88 BD 00 01
82B0- C9 45 F0 46 C9 2D 00 D0
82B8- B1 1A C9 2C F0 1C C9 30
82C0- D0 03 98 D0 E4 B1 1A C9
82C8- 30 F0 0F C9 20 F0 0B C9
82D0- 2A F0 07 C9 24 D0 D2 2C
82D8- 58 FF BD 00 01 91 1A CA
82E0- 10 C7 98 F0 1E 88 B1 1A
82E8- 50 03 A9 24 B8 C9 2C D0
82F0- 04 C8 B1 1D 88 91 1A 38
82F8- B0 E8 A9 23 A4 19 88 91
8300- 1A D0 FB A9 19 A0 00 85
8308- A0 84 A1 20 3D DB A5 19
8310- 20 25 E6 20 B7 00 C9 2C
8318- D0 03 4C 3E 82 C9 3B F0
8320- 03 4C FB DA 4C B1 00 20
8328- 7B DD 20 6C DD A5 A1 48
8330- A5 A0 48 20 BE DE 20 67
8338- DD 20 FB E6 68 85 A0 68
8340- 85 A1 8A F0 0B 8D 0A 86
8348- 20 3D DB CE 0A 86 D0 F8
8350- 20 B7 00 38 B0 C7 20 E3
8358- DF 85 85 84 86 A5 81 48
8360- AF 82 48 20 BE DE 20 E3
8368- DF 68 45 82 30 1F 68 45
8370- A1 30 1A A0 02 24 81 30
8378- 10 24 82 30 02 A0 04 B1
8380- 85 48 B1 83 91 85 68 91
8388- 83 88 10 F3 60 4C 76 DD
8390- 20 D9 F7 A0 02 18 A5 98
8398- 85 42 71 98 85 3C A5 9C
83A0- 85 43 C8 85 3D 38 A5 6D
83A8- 85 3E 88 F1 9B 85 6D A5
83B0- 6E 85 3F C8 F1 9B 85 6E
83B8- A0 00 4C 2C FE 20 46 E7
83C0- 86 51 A4 50 98 AA CA D0
83C8- FD 2C 30 C0 45 51 AA CA
83D0- 00 FD 2C 30 C0 88 D0 EC
83D8- 60 20 67 DD 20 FB E6 8A
83E0- 48 20 B7 00 C9 C5 D0 11
83E8- 20 B1 00 20 46 E7 8A A8
83F0- 20 C1 FB A4 50 68 91 28
83F8- 60 A2 35 4C 12 D4 20 67
8400- DD 20 FB E6 A4 A1 4C E4
8408- FB 20 67 DD 20 52 E7 A6
8410- 50 A9 C7 20 A8 FC CA D0
8418- F8 C6 51 30 04 A2 FF D0
8420- F0 60 D0 EC 60 20 46 E7
8428- 86 51 2C 30 C0 88 D0 04
8430- C6 51 F0 08 CA D0 F6 A6
8438- 50 4C 2A 84 20 B7 00 C9
8440- 3B D0 05 20 B1 90 D0 DD
8448- 60 A9 01 85 18 A9 04 85
8450- 1F 06 18 F0 10 A5 18 85
8458- 50 E6 1F A6 1F 86 51 20
8460- 2A 84 4C 51 84 60 20 7B
8468- DD 20 52 E7 20 1A D6 90
8470- 03 4C 41 D9 A2 5A 4C 12
8478- D4 20 7B DD 20 52 E7 20
8480- 1A D6 90 F0 A9 03 20 D6
8488- D3 A5 B9 48 A5 88 48 A5
8490- 76 48 A5 75 48 A9 R0 48
8498- 20 B7 00 20 41 D9 4C D2
84A0- D7 20 B7 00 F0 1A C9 22
84A8- F0 17 20 67 DD 20 FB E6
84B0- 8A 20 D2 84 20 B7 00 C9
84B8- 2C D0 05 20 B1 00 D0 E1
84C0- 60 20 B1 00 F0 FA C9 22
84C8- F0 D7 29 7F 20 D2 84 4C
84D0- C1 84 2C C1 C1 30 FB 8D
84D8- 90 C0 60 A9 00 48 48 4A
84E0- 29 03 09 04 85 51 68 29
84E8- 18 90 02 69 7F 85 50 0A

```

```

84F0- 0A 05 50 85 50 A0 00 B1
84F8- 50 30 08 29 3F C9 20 B0
8500- 02 09 40 49 80 20 D2 84
8508- C8 C0 28 D0 EA A9 0A 20
8510- D2 84 A9 0D 20 D2 84 68
8518- A8 C8 98 C9 18 D0 BE 60
8520- 20 67 DD 20 FB E6 8A 20
8528- 37 85 A0 00 91 50 88 D0
8530- FB E6 51 CA D0 F4 60 A2
8538- 00 86 50 A6 E6 00 40 F0
8540- 02 A2 20 86 51 A2 20 60
8548- 20 37 85 A0 00 B1 50 49
8550- FF 91 50 88 D0 F7 E6 51
8558- CA D0 F0 60 20 67 DD 20
8560- 52 E7 A5 50 85 3C A5 51
8568- 85 3D 20 BE DE 20 67 DD
8570- 20 52 E7 A5 50 85 3E A5
8578- 51 85 3F A9 C1 20 C0 DE
8580- 20 67 DD 20 52 E7 A5 50
8588- 85 42 A5 51 85 43 4C 2C
8590- FE AD EA B7 8D 00 86 AD
8598- E9 B7 8D FF 85 AD FD 85
85A0- C9 04 B0 05 A2 35 4C 12
85A8- D4 8D 07 86 A9 00 8D 06
85B0- 86 85 06 60 20 46 E7 E0
85B8- 10 B0 E9 8E 03 86 A6 50
85C0- E0 23 B0 E0 8E 02 86 20
85C8- 91 85 60 20 84 85 A9 01
85D0- AD 0A 86 A9 85 A0 FE 20
85D8- B5 B7 90 07 A9 80 85 06
85E0- 4C DD FB 60 20 B4 85 A9
85E8- 02 8D 0A 86 4C D3 85 20
85F0- 67 DD 20 FB E6 E0 04 90
85F8- AB 8E FD 85 60 96 01 60
8600- 01 00 12 06 0F 86 00 96
8608- 00 00 00 00 00 60 01 00
8610- 01 EF D8 20 67 DD 20 52
8618- E7 20 1A D6 A2 5A 90 20
8620- A0 04 B1 9B C9 83 D0 16
8628- A5 50 85 7B A5 51 85 7C
8630- 18 A5 9B 69 04 85 7D A5
8638- 9C 69 00 85 7E 60 A9 2A
8640- 4C 12 D4 4C 56 86

```

Super PRINT de D. Sureau -
Pom's 9 codes correspondant aux
lignes 55 à 181 du source original,
telogés à \$8646

*86FD,886D

```

86FD- 20 7B DD
8700- 20 6C DD A9 C5 20 C0 DE
8708- A5 25 48 A5 24 48 A5 A0
8710- 48 A5 A1 48 20 46 E7 8A
8718- F0 27 E0 19 B0 23 CA 86
8720- 25 20 22 FC A6 50 F0 19
8728- E0 29 B0 15 CA 86 24 68
8730- 85 A1 68 85 A0 20 3D DB
8738- 68 85 24 68 85 25 4C 22
8740- FC 68 68 68 85 24 68 85
8748- 25 20 22 FC A2 35 4C 12
8750- D4 2C 10 C0 20 0C FD 29
8758- 7F A8 A9 00 20 F2 E2 20
8760- D1 8A AA 4C 2B E8 68 87
8768- D0 08 C9 18 B0 07 AA C8
8770- 68 48 A9 00 8D 71 87 20
8778- 67 DD 20 52 E7 A5 50 8D
8780- 67 87 8D 6A 87 A5 51 8D
8788- 68 87 8D 6B 87 20 BE DE
8790- 20 67 DD 20 FB E6 8E 69
8798- 87 8E 6C 87 20 BE DE 20
87A0- 67 DD 20 FB E6 E0 00 F0
87A8- A3 8E 66 87 8E 70 87 A9
87B0- 00 8D 6E 87 8D 6D 87 8D
87B8- 6F 87 20 E7 87 18 AD 6D
87C0- 87 6D 70 87 8D 6D 87 90

```


87C8- 06 EE 6E 87 20 E7 87 38
 87D0- AD 6F 87 ED 6E 87 8D 6F
 87D8- 87 B0 E2 CE 70 87 20 E7
 87E0- 87 AD 70 87 D0 D7 60 AD
 87E8- 69 87 18 6D 6E 87 80 07
 87F0- C9 C0 80 03 20 00 88 38
 87F8- AD 69 87 ED 6E 87 90 4B
 8800- 48 AD 67 87 18 6D 70 87
 8808- B0 12 AC 68 87 F0 04 C9
 8810- 18 B0 19 AA 68 48 20 4C
 8818- 88 4C 2C 88 AC 68 87 D0
 8820- 08 C9 18 B0 07 AA C8 68
 8828- 48 20 4C 88 38 AD 67 87
 8830- ED 70 87 90 08 AC 68 87
 8838- AA 68 20 4C 88 4C 4B 88
 8840- AA 68 AC 68 87 F0 04 88
 8848- 20 4C 88 60 20 57 F4 AD
 8850- 71 87 F0 F7 AD 6A 87 AE
 8858- 6B 87 AC 6C 87 4C 3A F5
 8860- A9 A0 8D 71 87 4C 77 87
 8868- 20 B7 00 D0 01 60

ROUTINE VARIABLE de la Tortue

*8AD1.8AE3

8AD1- A9 80 85 14 20 E3 DF
 8AD8- 24 11 30 05 24 12 30 01
 8AE0- 60 4C 76 D0

Tortue de DUMA - Pom's 6 à
l'adresse \$886E - avec modifications
au-dessous.

*8C75.90D2

8C75- A9 00 8D
 8C78- CA 8C 20 7B DD 20 6C DD
 8C80- A5 A1 A0 02 B1 A0 99 18
 8C88- 00 88 10 F8 E6 18 A2 00
 8C90- C8 B1 19 09 80 9D 00 02
 8C98- C8 E8 C4 18 90 F3 CA AD
 8CA0- CA 8C D0 16 A9 A0 8D CA
 8CA8- 8C A9 8C 85 1A A9 C1 85
 8CB0- 19 A9 08 85 18 A0 00 4C
 8CB8- 91 8C A9 00 85 48 4C 70
 8CC0- FF A0 CE A0 C4 B7 C4 B2
 8CC8- C7 A0 00 20 91 90 A5 37
 8CD0- C9 8C F0 13 A9 8C 85 37
 8CD8- A9 F0 85 36 A9 8E 85 39
 8CE0- A9 32 85 38 20 EA 03 A9
 8CE8- 2C 8D F0 8C 8D 32 8E 60
 8CF0- 2C F0 FD 8D 88 8D 20 4A
 8CF8- FF 20 9C 8D A2 00 8E 8A
 8D00- 8D A6 32 30 05 A9 7F 8D
 8D08- 8A 8D AD 88 8D 30 0D A2
 8D10- 7F 8E 8A 8D 29 3F C9 20
 8D18- B0 02 09 C0 C9 A0 90 34
 8D20- 20 A8 8D A2 08 86 1F A5
 8D28- 25 0A 0A 0A 8D 89 8D AD
 8D30- 89 8D 20 B7 90 A5 24 18
 8D38- 65 20 A8 AD 00 20 4D 8A
 8D40- 8D 91 26 91 26 EE 89 8D
 8D48- EE 3C 8D D0 03 EE 3D 8D
 8D50- C6 1F D0 8D 8D 88 8D A6
 8D58- 23 CA 8E 89 8D A6 25 EC
 8D60- 89 8D 90 1E C9 8D F0 29
 8D68- C9 8A F0 25 A6 24 EC 8B
 8D70- 8D 90 0F C9 95 F0 1A AA

8D78- 10 17 C9 A0 B0 13 29 7F
 8D80- C9 20 20 3F FF 4C F0 FD
 8D88- 8D 17 00 27 91 11 31 51
 8D90- B1 A9 00 8D 8D 90 20 0B
 8D98- 90 4C 82 8D A5 21 18 65
 8DA0- 20 8D 88 8D CE 8B 8D 60
 8DA8- 38 E9 20 29 7F 8D 3C 8D
 8DB0- A9 00 8D 3D 8D 18 2E 3C
 8DB8- 8D 2E 3D 8D 2E 3C 8D 2E
 8DC0- 3D 8D 2E 3C 8D 2E 3D 8D
 8DC8- A5 CE 6D 3C 8D 8D 3C 8D
 8DD0- A5 CF 6D 3D 8D 8D 3D 8D
 8DD8- 60 20 B7 00 C9 28 D0 37
 8DE0- 20 67 DD 20 FB E6 8A 20
 8DE8- A8 8D AD 3C 8D 85 50 AD
 8DF0- 3D 8D 85 51 20 87 00 C9
 8DF8- 23 D0 1C A9 00 85 1F 20
 8E00- B1 00 20 67 DD 20 FB E6
 8E08- 8A A4 1F E6 1F 91 50 20
 8E10- B7 00 C9 2C F0 E9 60 A2
 8E18- 10 4C 12 D4 20 67 DD 20
 8E20- FB E6 E0 05 B0 07 BD 8C
 8E28- 8D 8D 41 8D 60 A2 35 4C
 8E30- 12 D4 2C 1B FD 20 4A FF
 8E38- A5 25 0A 0A 0A 18 69 07
 8E40- 20 B7 90 A5 24 18 65 20
 8E48- A8 B1 26 48 B1 28 85 1F
 8E50- 48 A9 FF 91 26 A9 FF 51
 8E58- 26 91 26 E6 4E D0 0F E6
 8E60- 4F 51 26 91 26 B1 28 AA
 8E68- A5 1F 91 28 86 1F 2C 00
 8E70- C0 10 E8 68 91 28 68 91
 8E78- 26 AD 00 C0 C9 8D D0 03
 8E80- 20 92 8E 20 3F FF 4C 9E
 8E88- 92 A9 4C 8D F0 8C 8D 32
 8E90- 8E 60 20 9C 8D A5 25 0A
 8E98- 0A 0A 85 1F C6 1F A5 24
 8EA0- 18 65 20 8D 90 90 CE 90
 8EA8- 90 A2 08 E6 1F A5 1F 20
 8EB0- B7 90 CA 30 0F AC 8B 8D
 8EB8- A9 00 91 26 88 CC 90 90
 8EC0- D0 F8 F0 E7 60 A6 23 8E
 8EC8- 89 8D A9 00 85 24 8A 85
 8ED0- 25 C6 23 20 92 8E A5 25
 8ED8- C5 22 D0 F5 60 20 46 E7
 8EE0- 8E 8D 90 20 9C 8D A5 50
 8EE8- C9 01 F0 17 C9 02 F0 50
 8EF0- C9 03 D0 03 4C 83 8F C9
 8EF8- 04 D0 03 4C 08 90 A2 35
 8F00- 4C 12 D4 A5 23 0A 0A 0A
 8F08- 85 1F 20 9C 8D A5 22 0A
 8F10- 0A 0A 8D 90 90 C6 1F A5
 8F18- 1F 20 B7 90 A4 20 C8 B1
 8F20- 26 88 91 26 C8 C8 CC 8B
 8F28- 8D 90 F4 A5 1F 20 B7 90
 8F30- AD 8D 90 AC 8B 8D 91 26
 8F38- A5 1F CD 90 90 D0 D6 60
 8F40- A5 23 0A 0A 0A 85 1F 20
 8F48- 9C 8D A5 22 0A 0A 0A 8D
 8F50- 90 90 A4 20 88 8C 89 8D
 8F58- C6 1F A5 1F 20 D7 90 AC
 8F60- 8B 8D 88 B1 26 C8 91 26
 8F68- 88 88 CC 89 8D D0 F4 A5
 8F70- 1F 20 B7 90 AD AD 90 A4
 8F78- 20 91 26 A5 1F CD 90 90
 8F80- D0 D6 60 A5 23 0A 0A 0A
 8F88- 8D 8F 90 38 E9 08 8D 8E
 8F90- 90 A5 22 0A 0A 8D 90
 8F98- 90 20 9C 8D A4 20 88 8C
 8FA0- 89 8D AD 8E 90 38 E9 01
 8FA8- CD 90 90 F0 34 8D 8E 90
 8FB0- 20 B7 90 A5 26 85 06 A5
 8FB8- 27 85 07 AD 8F 90 38 E9
 8FC0- 01 8D 8F 90 AD 8F 90 20
 8FC8- B7 90 A5 26 85 08 A5 27
 8FD0- 85 09 AC 88 8D B1 06 91
 8FD8- 08 88 CC 89 8D D0 F6 F0
 8FE0- C1 AD 8F 90 CD 90 90 F0
 8FE8- 21 38 E9 01 8D 8F 90 20
 8FF0- B7 90 A5 26 85 08 A5 27
 8FF8- 85 09 AD 8D 90 AC 8B 8D

9000- 91 08 88 CC 89 8D D0 F8
 9008- F0 D7 60 A5 22 0A 0A 0A
 9010- 8D 8F 90 CE 8F 90 18 69
 9018- 07 8D 8E 90 A5 23 0A 0A
 9020- 0A 8D 90 90 20 9C 8D AD
 9028- 8E 90 18 69 01 CD 90 90
 9030- B0 30 8D 8E 90 20 B7 90
 9038- A5 26 85 06 A5 27 85 07
 9040- AD 8F 90 18 69 01 8D 8F
 9048- 90 20 B7 90 A5 26 85 08
 9050- A5 27 85 09 AC 8B 8D B1
 9058- 06 91 08 C4 20 F0 C8 88
 9060- 10 F5 CE 90 90 AD 8F 90
 9068- CD 90 90 F0 9D 18 69 01
 9070- 8D 8F 90 20 B7 90 A5 26
 9078- 85 08 A5 27 85 09 AD 8D
 9080- 90 AC 8B 8D 91 08 C4 20
 9088- F0 DB 88 10 F7 00 00 00
 9090- BF A9 C0 85 1F A5 E6 48
 9098- A9 00 85 E6 C6 1F A5 1F
 90A0- 20 11 F4 A4 1F A5 27 99
 90A8- D3 90 A5 26 99 93 91 A5
 90B0- 1F D0 E9 68 85 E6 60 A8
 90B8- 20 C8 90 B9 93 91 85 26
 90C0- B9 D3 90 05 E6 85 27 60
 90C8- A5 E6 C9 40 F0 04 A9 20
 90D0- 85 E6 60

Table

*9253.94F5

9253- 20 67 DD 20 52
 9258- E7 A5 50 85 06 A5 51 85
 9260- 07 20 BE DE 20 67 DD 20
 9268- 52 E7 A0 00 A5 50 91 06
 9270- A5 51 C8 91 06 60 A9 4C
 9278- 85 0A A9 83 85 0B A9 92
 9280- 85 0C 60 20 52 E7 A0 01
 9288- B1 50 A8 88 B1 50 A8 8A
 9290- 4C F2 E2 A9 9E 85 38 A9
 9298- 92 85 39 4C EA 03 20 1B
 92A0- FD 85 06 AC 61 C0 30 06
 92A8- AC 62 C0 30 01 60 86 1F
 92B0- A0 18 20 E4 FB A0 00 A6
 92B8- 1F A5 06 C9 FF F0 4B C9
 92C0- B0 90 0E C9 B8 B0 0A 38
 92C8- E9 B0 A8 99 50 C0 4C 9E
 92D0- 92 B9 15 93 F0 31 C5 06
 92D8- F0 06 C8 C8 C8 4C D1 92
 92E0- C8 B9 15 93 85 18 C8 B9
 92E8- 15 93 85 19 A0 00 B1 18
 92F0- F0 0F C9 8D F0 0E 9D 00
 92F8- 02 E8 C8 20 ED FD 4C EE
 9300- 92 A9 A0 60 4C 8E FD A9
 9308- 20 60 20 10 FC A9 A0 20
 9310- ED FD A9 88 60 D0 8B 93
 9318- C3 91 93 C5 9A 93 AA AE
 9320- 93 D4 88 93 C9 BE 93 DB
 9328- C4 93 C0 CA 93 CA D1 93
 9330- DD DE 93 D5 E5 93 C8 F2
 9338- 93 C7 F7 93 A1 FB 93 CC
 9340- 08 94 D3 0D 94 C2 12 94
 9348- BD 17 94 AE 1F 94 D2 25
 9350- 94 D1 29 94 DC 2E 94 CD
 9358- 33 94 A4 39 94 BC 3F 94
 9360- BE A5 94 CB 4A 94 CE 58
 9368- 94 D9 5D 94 DE 62 94 A5
 9370- 86 94 89 AA 94 A3 B3 94
 9378- DA BD 94 D8 C6 94 D7 D0
 9380- 94 AF D8 94 BF E3 94 A7
 9388- ED 94 00 D0 D2 C9 CE D4
 9390- 00 C3 C1 D4 C1 CC CF C7
 9398- 8D 00 C8 CF CD C5 BA D0
 93A0- CF CB C5 B3 B3 AC B2 B8
 93A8- BA CC C9 D3 D4 00 C3 C1
 93B0- CC CC AD B1 B5 B1 8D 00
 93B8- A6 A0 C4 CF DB 00 C9 CE
 93C0- D0 D5 D4 00 A6 D0 CC C1
 93C8- D9 00 C1 D3 D3 C5 CD 8D

```

93D0- 00 A6 D3 D5 D0 C5 D2 A0
93D8- D0 D2 C9 CE D4 00 A6 CE
93E0- CF C9 D3 C5 00 A6 D0 D2
93E8- C9 CE D4 A0 D5 D3 CE C7
93F0- BE 00 C8 C7 D2 8D 00 C7
93F8- D2 8D 00 D4 C5 D8 D4 BA
9400- CE CF D2 CD C1 CC 8D 00
9408- CC CF C1 C4 00 D3 C1 D6
9410- C5 00 C2 D2 D5 CE 00 C3
9418- CF CC CF D2 A0 BD 00 C8
9420- D0 CC CF D4 00 D2 D5 CE
9428- 00 A6 C7 C5 D4 00 CC C9
9430- D3 D4 00 CD C9 C4 A4 A8

```

```

9438- 00 C3 C8 D2 A4 A8 00 D0
9440- C5 C5 CB A8 00 D0 CF CB
9448- C5 00 D0 CF CB C5 AD B1
9450- B6 B3 B6 B8 AC B0 BA 00
9458- CE C5 D8 D4 00 C3 C1 CC
9460- CC 00 BF A2 C8 C9 CD C5
9468- CD A0 BD A0 A2 D0 C5 C5
9470- CB A8 B1 B1 B5 A9 AB D0
9478- C5 C5 CB A8 B1 B1 B6 A9
9480- AA B2 B5 B6 8D 00 BF A2
9488- CC CF CD C5 CD A0 BD A0
9490- A2 D0 C5 C5 CB A8 B1 B0
9498- B5 A9 AB D0 C5 C5 CB A8

```

```

94A0- B1 B0 B6 A9 AA B2 B5 B6
94A8- 8D 00 A0 A0 A0 A0 A0 A0
94B0- A0 A0 00 A0 CE A0 C5 C4
94B8- B2 B4 C7 8D 00 A6 C3 C9
94C0- D2 C3 CC C5 BD 00 A6 D8
94C8- C3 C9 D2 C3 CC C5 BD 00
94D0- A6 A0 D7 C1 C9 D4 A0 00
94D8- D4 C5 D8 D4 BA C8 CF CD
94E0- C5 8D 00 A0 D2 CE C4 A8
94E8- B1 A9 AA A0 00 A6 D0 D2
94F0- CF D5 D4 A0 00 00

```

* Remplacer "suite Instruction" original par celui-ci.
 * Supprimer les lignes 65 à 90 de l'original.
 * Assembler à l'adresse \$886E

```

1 *****
2
3 * Suite instructions
4
5 SORTIE JSR $87
6       CMP #'J'
7       BNE SSUITE

```

```

8       JSR $B1
9       RTS
10 SSUITE LDA #' ;'
11       JSR $DECO
12       JMP $8868
13
14 *****

```

*7000,7300

```

7000- 00 00 00 00 00 00 00 00
7008- 1C 1C 1C 1C 1C 00 1C 00
7010- 36 36 24 12 00 00 00 00
7018- 22 7F 22 22 22 7F 22 00
7020- 08 3E 02 3E 20 3E 08 00
7028- 47 25 17 08 74 52 71 00
7030- 0C 12 02 04 0A 12 2E 00
7038- 1C 1C 10 0C 00 00 00 00
7040- 18 0C 06 06 06 0C 18 00
7048- 06 0C 18 18 18 0C 06 00
7050- 84 95 8E 84 8E 95 84 80
7058- 00 08 08 3E 18 18 00 00
7060- 00 00 00 00 1C 1C 10 0C
7068- 00 00 00 3E 3E 00 00 00
7070- 00 00 00 00 1C 1C 1C 00
7078- 40 20 10 08 04 02 01 00
7080- 1F 11 19 15 13 13 1F 00
7088- 07 04 04 04 04 04 1F 00
7090- 3F 20 20 3F 03 03 3F 00
7098- 1E 10 10 1E 18 18 1F 00
70A0- 91 91 91 91 8F 98 98 80
70A8- 1F 01 01 1F 18 19 1F 00
70B0- 9F 91 81 9F 93 93 9F 80
70B8- 9F 91 88 84 86 86 86 80
70C0- 1F 11 11 1F 13 13 1F 00
70C8- 1F 11 11 1F 18 19 1F 00
70D0- 00 1C 1C 00 1C 1C 00 00
70D8- 00 1C 1C 00 1C 1C 10 0C
70E0- 10 08 04 03 04 08 10 00
70E8- 00 3E 3E 00 3E 3E 00 00
70F0- 01 02 04 18 04 02 01 00

```

```

70F8- 7E 42 40 78 08 00 08 00
7100- 3C 24 3C 00 00 00 00 00
7108- 1F 11 3F 21 23 23 23 00
7110- 1F 11 3F 21 23 23 3F 00
7118- 3F 21 01 03 03 23 3F 00
7120- 1F 21 21 23 23 23 1F 00
7128- 3F 01 0F 01 03 03 3F 00
7130- 3F 01 1F 01 03 03 03 00
7138- 3F 21 01 39 23 23 3F 00
7140- 21 21 3F 21 23 23 23 00
7148- 04 04 04 0C 0C 0C 0C 00
7150- 20 20 20 30 30 31 1E 00
7158- 21 11 09 0F 13 23 23 00
7160- 01 01 01 03 03 03 3F 00
7168- 3F 49 49 49 4B 4B 4B 00
7170- 27 29 29 29 2B 2B 33 00
7178- 3F 21 21 21 23 23 3F 00
7180- 3F 21 21 3F 03 03 03 00
7188- 3F 21 21 29 2B 13 2F 00
7190- 3F 21 21 3F 13 23 23 00
7198- 3F 01 01 3F 30 30 3F 00
71A0- 3F 04 04 04 0C 0C 0C 00
71A8- 21 21 21 23 23 23 3F 00
71B0- 21 21 23 23 22 14 08 00
71B8- 41 41 49 49 4B 4B 7F 00
71C0- 21 12 0C 0C 12 21 21 00
71C8- 21 21 21 3F 04 0C 0C 00
71D0- 3F 10 08 04 02 03 3F 00
71D8- 1E 02 02 02 02 02 1E 00
71E0- 01 02 04 08 10 20 40 00
71E8- 3C 20 20 20 20 20 3C 00
71F0- 00 08 1C 36 63 00 00 00
71F8- 00 00 00 00 00 00 00 00

```

```

7200- 00 00 00 00 00 00 00 00
7208- 00 00 3E 20 3E 22 3E 00
7210- 02 02 3E 22 22 22 3E 00
7218- 00 00 3E 02 02 02 3E 00
7220- 20 20 3E 22 22 22 3E 00
7228- 00 00 3E 22 3E 02 3E 00
7230- 3C 04 1E 04 04 04 04 00
7238- 00 00 3E 22 22 3E 20 3E
7240- 02 02 3E 22 22 22 22 00
7248- 18 00 18 18 18 18 18 00
7250- 00 10 00 10 10 10 12 1E
7258- 02 02 12 0A 06 0A 12 00
7260- 18 18 18 18 18 18 18 00
7268- 00 00 7F 49 49 49 49 00
7270- 00 00 3E 22 22 22 22 00
7278- 00 00 3E 22 22 22 3E 00
7280- 00 00 3E 22 22 3E 02 02
7288- 00 00 3E 22 22 3E 20 20
7290- 00 00 3E 22 02 02 02 00
7298- 00 00 3E 02 3E 20 3E 00
72A0- 00 04 3E 04 04 04 3C 00
72A8- 00 00 24 24 24 24 3C 00
72B0- 00 00 22 22 22 14 08 00
72B8- 00 00 41 49 49 49 7F 00
72C0- 00 00 22 14 08 14 22 00
72C8- 00 00 22 22 22 3E 20 3E
72D0- 00 00 3E 10 08 04 3E 00
72D8- 00 00 00 00 00 00 00 00
72E0- 00 08 08 08 08 08 08 08
72E8- 00 00 00 00 00 00 00 00
72F0- 00 00 00 00 00 00 00 00
72F8- 00 00 00 00 00 00 00 00
7300- FF

```

Ecriture entrelacée/verticale

Voici deux petites routines qui se suffisent à elles-mêmes et peuvent éventuellement être utilisées pour une présentation plus originale de vos programmes.

Marcel Cottini

```

5 TEXT : HOME
10 FOR I = 1 TO 7: POKE 32,2 * I - 1: P
   OKE 33,1: HOME : READ A$: PRINT A
   *: NEXT I: TEXT : VTAB 23: END

```

20 DATA LECTURE VISUELLEMENT,EN COLON
 NES,VERTICALES,MIS AU POINT,PAR C
 OTTINI MARCEL,POUR DEMONSTRATION.

```

10 TEXT : HOME
320 A1$ = "Normalement,l'écriture s'atti
  che, "
330 A2$ = "du haut vers le bas,et non de
  droite, "
340 A3$ = "vers la gauche ou inversement
  "
350 A4$ = "Ceci n'est que démonstratif,e
  t arrangé à "
360 A5$ = "cet effet. "
370 A6$ = "Les lignes s'entrelacent de g
  auche à "
380 A7$ = "droite,et inversement. "

```

```

390 FOR I = 1 TO 40
400 VTAB 7: HTAB 41 - I: PRINT LEFT$ (
  A1$,I)
410 VTAB 8: HTAB 1: PRINT RIGHT$ (A2$,
  I)
420 NEXT
430 FOR I = 1 TO 40
440 VTAB 9: HTAB 41 - I: PRINT LEFT$ (
  A3$,I)
450 VTAB 11: HTAB 1: PRINT RIGHT$ (A4$,
  I)
460 VTAB 12: HTAB 41 - I: PRINT LEFT$
  (A5$,I)
470 NEXT
480 FOR I = 1 TO 40
490 VTAB 14: HTAB 1: PRINT RIGHT$ (A6$,
  I)
500 VTAB 15: HTAB 41 - I: PRINT LEFT$
  (A7$,I)
500 NEXT

```


Visicalc Advanced Version (VAV) est déjà sorti depuis quelque temps en version //e, mais je n'avais pas encore eu l'occasion de le tester et, partant, de vous en parler. VAV se présente sous la forme d'un classeur avec une documentation de plus de 200 pages, une disquette de boot et une disquette programme : présentation professionnelle. Il est indispensable de disposer de 128K de mémoire, et très souhaitable d'avoir deux lecteurs de disquettes; il faut enfin pouvoir dépenser 3.200 F HT prix public.

Vitesse de calcul

En quelques mots, c'est un tableur presque aussi rapide que Visicalc et plus puissant que Multiplan selon de nombreux critères. Ainsi, j'ai comparé diverses mesures de temps et de place mémoire pour le tableau de trésorerie utilisé dans les livres "Visicalc sur Apple" et "Multiplan pour Apple"; en voici les résultats :

	Visicalc	Multiplan	Magicalc	V.A.V.
Chargement	60"	15"	53"	115"
Mémoire libre	21K	25K	76K	56K
Temps de calcul	23"	50"	25"	33"

Un autre tableau très simple a été constitué, avec des chiffres dans les dix premières cellules de la colonne 1, puis 10 colonnes de 100 lignes effectuant la somme des dix valeurs de la colonne 1. Le tableau suivant est alors obtenu :

	Visicalc	Multiplan	Magicalc	V.A.V.
Mémoire libre	15K	18K	64K	50K
Temps de calcul	23"	190"	40"	50"

On constate que le classement selon la vitesse de calcul reste le même, avec toutefois des rapports de vitesse différents. Ici, les différences de vitesse sont amplifiées, ce qui est normal pour un tableau plus grand.

Visicalc et VAV ont un chargement sensiblement plus long, car celui-ci requiert un recalcul complet, contrairement au cas de Multiplan. Le chargement de VAV est sensiblement plus long, ainsi que l'appel de nombreuses commandes. En effet, VAV ne réside pas tout entier en mémoire, ce qui a l'avantage de laisser à l'utilisateur 70K, contrairement aux 35K de Multiplan et aux 34K de Visicalc : seul Magicalc fait mieux avec 90K. On peut donc utiliser avec VAV des tableaux deux fois plus importants. Cela se paye par la nécessité de garder la disquette programme dans le premier lecteur (il faut donc en avoir deux) et les va-et-vient ayant lieu lors des appels de commandes.

Comparons à présent VAV à Multiplan, plutôt qu'à son ancêtre Visicalc, qui date franchement (ce qui a fait le succès de Magicalc, qui en est une amélioration sensible). Par rapport à Multiplan, VAV apporte de nouvelles

Visicalc Advanced Version

Hervé Thiriez

possibilités de formatage, des fonctions nouvelles et, surtout, la préprogrammation de séquences de touches.

Formatage

Comme dans Multiplan, le formatage comporte deux parties, l'une relative au format proprement dit (le "code format" de Multiplan), l'autre au cadrage, deux domaines dans lesquels les options de VAV sont plus nombreuses. Dans VAV, le format (/F pour le format et /A pour les attributs) sert aussi à définir le type d'information autorisée, son cadrage dans la cellule (plus riche que pour Multiplan) et le fait de cacher (/AH pour Hide) ou non son contenu. L'utilisateur peut définir des formats répétitifs qui seront alors appelés directement avec "/F=x", où x est le code du caractère associé au format personnalisé à utiliser. Il y a 37 commandes débutant par "/A" et neuf par "/F". Les formats, comme les attributs, sont reproductibles séparément du contenu des cellules.

Fonctions

Par rapport aux fonctions de Multiplan, il manque dans VAV les fonctions DELTA et NBITER (liées au calcul itératif, et donc inutiles dans VAV), ECARTYPE, STXT, SIGNE, LIGNE et COLONNE.

Les fonctions nouvelles sont principalement les fonctions financières, les fonctions de gestion de dates et DOTPROD. Cette dernière fonction, introuvable dans les autres tableurs classiques (même Lotus 1-2-3...), effectue la somme des produits de deux groupes de valeurs, le produit scalaire.

Les fonctions financières sont FV (valeur future), PERIODS (calcul du nombre de périodes de remboursement), PMT (mensualités à payer), et IRR (taux de rendement interne), en plus du désormais classique NPV (la valeur actuelle nette).

Il y a neuf fonctions de calcul de dates, permettant entre autres de calculer le nombre de jours entre deux dates.

Préprogrammation

On peut, à l'aide de "/K=", programmer des séquences de touches complètes, appelées simplement par "Ctrl-S x", où x sera le code de commande de la séquence souhaitée. Cette capacité de préprogram-

mer des traitements standards ne se trouve actuellement que dans Lotus 1-2-3 : elle n'est disponible dans aucun tableur actuel sur Apple II.

Cette préprogrammation permet de gagner beaucoup de temps lors de la réalisation de tableaux, en évitant les opérations répétitives, telles que l'insertion d'une colonne de séparation avec des "!", l'insertion d'une ligne de titrage avec les mois de l'année, ou la création d'une formule de totalisation en ligne puis sa répétition vers le bas.

La préprogrammation permet aussi par exemple d'automatiser des opérations dans lesquelles il est important d'éviter les erreurs, telles que la lecture ou la sauvegarde partielle d'un tableau en format DIF.

Limites de VAV

VAV est inférieur à Multiplan au niveau de la "consolidation", du fenêtrage et de l'utilisation de noms. Il ne peut, comme Multiplan, lier deux ou plusieurs tableaux de façon active. Il est limité à deux fenêtres, plus les titres, ce qui fait tout de même cinq fenêtres par rapport aux huit (les titres en prennent quatre) de Multiplan. Il ne peut non plus utiliser des noms dans ses formules, pour définir par exemple une cellule de contribution par "Px Vente - Px Achat".

Conclusion

Comme je l'avais dit dans la conclusion de l'analyse de Magicalc, il est difficile de dire si un tableur est "meilleur" qu'un autre; cela n'a pas beaucoup de sens. Finalement, VAV donne à l'utilisateur une plus grande capacité mémoire et des calculs plus rapides que ceux de Multiplan; sur ces points, il est par contre inférieur à Magicalc. Ses fonctions financières sont sensiblement plus riches, et il est le seul des tableurs sur Apple à offrir la préprogrammation de séquences de touches et la fonction DOTPROD. En compensation, pas d'utilisation de noms ni de liens externes facilitant la consolidation. En conclusion, ce produit possède un excellent rapport performance/prix.

En ce qui concerne Multiplan, n'oublions pas la sortie (septembre 84) de la version 1.07, qui ajoute des fonctions financières (dont le taux de rendement interne), est compatible avec le //c et exploite les 64K de la carte 80 colonnes étendue. Quand je pense que j'ai déjà plus de 10 tableurs sur mon brave Apple II... ■

Il était temps qu'un capitaliste

MAO
TSE-TUNG

ENGELS

LENIN

KARL
MARX

TROTSKY

DAS KAPITAL

fasse une révolution.



N'apprenez plus à devenir une machine, Apple a inventé Macintosh...



PEACHTREE, LA COMP QUE C'EN EST PRESQU



La comptabilité, c'est long, compliqué, fastidieux !
Tout cela était vrai avant Mac Accounting. Avec Mac Accounting, le nouveau logiciel de comptabilité de Peachtree, finis les livres de caisse en deux exemplaires, finis les bons de caisse, finis les carbones, terminées les mises à jour qui font perdre du temps. Mac Accounting est un programme comptable développé spécialement par Peachtree pour le Mac Intosh. Et dans le domaine particulier de la comptabilité, la compétence de Peachtree n'est plus à prouver. Avec plus de 100.000 utilisateurs dans le monde entier, Peachtree possède une très large expérience des besoins des petites et moyennes entreprises en matière de comptabilité. Spécialement étudié pour un utilisateur n'ayant aucune connaissance infor-

matique préalable, Mac Accounting est conçu pour être le plus proche des systèmes manuels... avec l'exactitude et la rapidité des solutions informatiques. Un comptable reconnaît immédiatement les pages d'un journal de caisse ou du Grand Livre : c'est exactement ce qu'il retrouvera sur l'écran du Mac Intosh lorsqu'il utilisera Mac Accounting.

Mac Cash est le premier des deux programmes qui constituent Mac Accounting. C'est un livre de caisse sur écran. Idéal pour les PME, ses caractéristiques parlent d'elles-mêmes. La taille du journal, le titre des colonnes, la période comptable et les formats monétaires sont définis par l'utilisateur. Mac Cash accepte jusqu'à cinq taux de taxe, et huit colonnes d'analyse des prix hors-taxes, sa mise à jour est automatique au fur et à mesure de la

TA DEVIENT SI SIMPLE E UN PÈCHÉ.

alliance



saisie des opérations, enfin l'utilisateur spécifie lui-même les modèles des états de sortie et leur contenu. Bref, un outil simple et performant.

Mac Ledger est le second module de Mac Accounting, c'est la partie comptabilité générale. Le format du journal est standard et peut être utilisé aussi bien pour les comptes fournisseurs que pour les comptes clients. L'utilisateur définit lui-même le format des fiches comptables qui se sélectionnent facilement à l'aide de la souris Mac Intosh. Les opérations saisies peuvent être analysées par type de produit, par territoire, par vendeur ou par toute autre variable définie par l'utilisateur... Les créances sont analysées sur trois périodes, ce qui facilite contrôle et réajustement. En résumé le complément indispensable de Mac Cash.

Dernier avantage: Mac Accounting a été conçu en français et peut être utilisé par n'importe quel utilisateur en moins de trente minutes. Le doute n'est plus permis: avec Mac Intosh et Mac Accounting la comptabilité devient si simple que c'est presque un péché.

Peachtree est distribué par Logiciel PC: 113, bd Pèreire 75017 Paris. Tél.: 763.62.88.



Peachtree

simplement génial

avec les livres P.S.I.

Les essais du Macintosh

Hervé Thiriez

Disquettes

La Pierre Molle, Jeu réalisé par Bruno Rives et diffusé par ACI.

Bruno Rives est le responsable de la ligne 68000 chez Apple. Sédurir ce n'est donc pas le premier venu; il a même signé des articles dans notre revue, c'est tout dire !

La Pierre Molle est un jeu d'adresse nécessitant aussi un certain sens stratégique. Vous tombez en panne avec votre soucoupe volante sur une planète où se trouve heureusement un bidon de carburant. Votre travail consistera à aller chercher ce bidon en sautant sur des pierres, certaines dures, d'autres molles.

Ces pierres tombent du ciel et vous devez les éviter lors de leur chute, puis les utiliser. Ce jeu est une excellente démonstration de la capacité d'apprentissage de l'être humain. Au début, vous n'allez franchement pas loin... un beau jour, vous devenez capitaine, puis commandant et, si vous devenez très fort, chevalier.

Les effets visuels et sonores du jeu sont fort réussis : manifestement, l'Apple II est dépassé sur ces deux plans. Nous avons regretté, après un certain nombre de parties, que l'on ne puisse rendre la chute des pierres aléatoires. Autre regret, l'impossibilité de stocker le meilleur score de plusieurs joueurs simultanément. Par contre, nous avons apprécié la possibilité de choisir entre deux vitesses de jeu, y compris en cours de partie. En conclusion, il s'agit là d'un jeu addictif qui m'a, à plusieurs reprises, forcé à me coucher plus tard que prévu. Heureusement, il est possible de couper le son à loisir.

Eleugram, disquette mensuelle de jeu vendue par CompuSoft.

Nous avons été ahuris de voir que, dès le lancement effectif du Macintosh en France, on pouvait ainsi trouver une disquette mensuelle de jeu.

Il s'agit d'une disquette comportant une vingtaine de problèmes logiques pour lesquels une grille d'analyse est proposée sur l'écran. C'est le type de problème et de grille que l'on trouve dans la revue *Eleusis* (la similitude des noms n'est pas due au hasard).

Le programme fournit le cas échéant une assistance au joueur. Il est possible de faire remplir par le programme toutes les cases directement "impliquées" par les cases que vous noircissez à un moment donné.

Les problèmes sont variés, la présentation soignée et l'assistance au

joueur bien réglable. Mon seul regret, en tant que grand joueur, est qu'il n'y ait sur la disquette qu'un seul type de jeu, même si cela correspond à de nombreux problèmes. A moins d'être un fana de jeux logiques de ce type, on préférera probablement acheter une disquette de temps en temps plutôt que de s'abonner.

Matériel

Alpha Systèmes annonce des unités de disques durs pour le Macintosh : 5 Mo sur cartouche amovible, 10 Mo fixe, 5 Mo fixe + 5 Mo amovible et 10 Mo fixe + 5 Mo amovible. L'unité se connecte sur la sortie RS 432 du Mac, qui devra donc choisir entre communiquer et accéder au disque...

Enfin, ceux qui désirent avoir le processeur du Mac, le fameux 68000, sur leur Apple II+ ou IIe, seront comblés par la carte 68000 SAY-BROOK (avec mémoire de 128K) annoncée elle aussi par Alpha Systèmes : 8 MHz (9.950 F) ou 12,5 MHz (12.950 F) ou enfin 14 MHz (14.950 F). Les programmes développés en Applesoft tourneront 10 à 30 fois plus vite avec l'interpréteur de la carte; le Pascal p-UCSD version IV.1 étant lui aussi fourni en standard, on pourra tourner 5 à 20 fois plus vite en Pascal et développer des programmes destinés aux IBM PC/XT, Rainbow 100... à partir du bon vieux Apple II.

Bibliographie

Macintosh/Multiplan/MacPaint.

par Eddie Adamis, Cedic/Nathan, 142 pages, 89 F.

Réaliser un livre d'initiation à MacPaint et à Multiplan sur le Macintosh correspond à une gageure. En effet, la particularité de ce matériel est de rendre aisé tout apprentissage : à quoi peut alors servir un livre d'initiation ?

Il est certain qu'un tel livre n'est pas indispensable; il facilitera cependant le travail du débutant, en l'amenant à "réaliser une démonstration" livre à la main.

Les deux premiers tiers du livre sont consacrés à Multiplan. La démonstration est claire, mais on n'y trouvera ni astuces d'utilisation, ni exemples divers. C'est une bonne démonstration, et rien de plus. Le reste de l'ouvrage est une "démonstration" de MacPaint. En conclusion, le livre est clair et fa-

cile d'accès. Ceci dit, 45 minutes de présentation par un utilisateur averti vous en apprendront autant.

The Apple Macintosh Book, par Cary Lu, Microsoft Press, 385 pages, \$18.95.

Présentation : une couverture rouge traversée par un sigle "Mac", en rouge plus foncé et en forme de graffiti.

Le seul reproche que l'on puisse faire à ce livre, c'est qu'il ne soit pas encore disponible en français. Vendu par Microsoft (en France aussi), cet ouvrage est la bible de l'utilisateur du Mac. Après une brève section de présentation, l'auteur présente en 115 pages tous les logiciels tournant déjà ou prévus à court terme (y compris le Basic Apple, différent du Basic Microsoft). Une troisième section analyse le matériel, élément par élément : on trouve dans cette partie (136 pages) un chapitre sur le dépannage, un sur l'entretien, et même un chapitre de comparaison avec l'IBM PC. La dernière partie de l'ouvrage parle de communications, de la façon d'échanger des informations avec d'autres ordinateurs, montre comment photographier l'écran et enfin contient un chapitre technique plein de renseignements utiles.

Le livre est bien présenté : la table des matières détaillée permet de retrouver rapidement ce que l'on cherche. Cary Lu, éditeur micro-informatique de High Technology, profite de son expérience pour parler des développements futurs du Macintosh comme de la micro-informatique en général. Les sujets couverts sont multiples, il y a même quatre pages s'adressant aux handicapés et leur montrant comment utiliser au mieux le Macintosh.

En conclusion, ce livre est un must pour ceux qui souhaitent connaître (et maîtriser) toutes les possibilités du Macintosh. Feuilletez-le avant de l'acheter, vous serez convaincu.

Avis aux auteurs : nous ne manquons pas, loin de là, de contributions intéressantes pour la famille Apple II. Par contre, vu la nouveauté du sujet, nous n'avons pas encore reçu de contributions pour le Macintosh. Faites partager votre savoir-faire et envoyez-nous vos trucs et programmes.

**FAITES
UN CADEAU
A VOTRE APPLE®
POUR :**



169

F
(TVA incluse)

Comme des milliers
d'utilisateurs en France,
découvrez avec

BABY MEMDOS

l'univers fabuleux des possibilités de MEMDOS

- un gestionnaire d'écran
- un séquentiel indexé multiclé
- un gestionnaire d'erreurs
- la notion de sous-programmes avec passage de paramètres et beaucoup plus...

Pour Apple II, Iie, Iic et III.

Disquette et Documentation.



MEMDOS
Pomme d'Or du Meilleur Logiciel Système
décerné par APPLE

BON DE COMMANDE

A adresser à : **MEMSOFT S.A.** 3 rue Meyerbeer - 06000 Nice

- ☐ Je joins un chèque de 169 F + 15 F de port - Soit 184 F ☐ Contre remboursement 169 F + 30 F de port - Soit 199 F
à l'ordre de Memsoft S.A.

NOM..... SOCIETE.....

ADRESSE..... TEL.....

☐ Je suis Amateur ☐ Professionnel

Je possède un Apple type périphériques.....

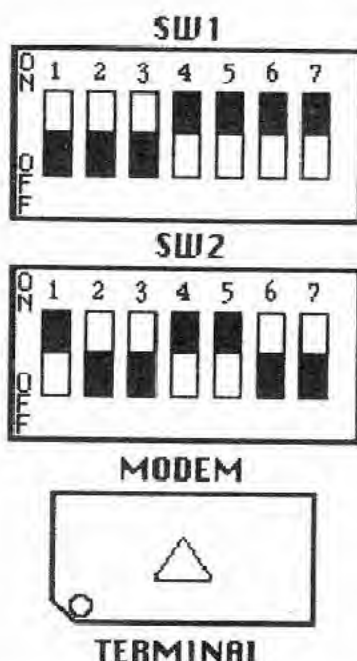
MacWrite vers AppleWriter IIe

Pom's

Vous utilisez maintenant un Macintosh et Macwrite pour vos textes mais, comme nous (voir Pom's 10, "La communication avec l'Apple"), vous avez mis au point des applications en WPL sur votre Apple IIe. Les deux petits programmes écrits en Basic ici proposés vous permettent de transmettre vos textes depuis le Mac vers l'Apple IIe, sans avoir recours à un logiciel de communication comme MacTerminal.

Pour utiliser ces programmes, il vous faut, outre le Mac et l'Apple IIe, une carte super série connectée dans le slot 2, et une disquette MS-Basic. Pour relier les deux appareils, on branche le câble normalement destiné à l'imprimante ImageWriter sur la carte super série, elle-même ainsi configurée :

Le texte écrit avec MacWrite doit être sauvegardé avec l'option "Le texte seul" et, si vous ne disposez que



d'un lecteur de disquettes, sur la disquette où se trouve le programme "émetteur". Ensuite, tout est très simple :

- le Macintosh vous demande le nom du fichier à transférer;
- le texte est transmis à l'Apple IIe;
- le IIe vous demande le nom sous lequel vous désirez conserver le fichier transmis, ainsi que le numéro du lecteur de destination;
- le texte est stocké sur disquette, prêt à être chargé avec AppleWriter.

Note : dans certains cas, la carte super série annule le code 127 utilisé pour marquer la fin du fichier. Si cela se produit, il vous suffit d'envoyer "manuellement" ce code pour que tout rende dans l'ordre (LPRINT CHR\$(127)).

RECUP APPLE II

```

1 PRINT CHR$(4)"BLOAD TEST
5 TEXT : HOME
7 REM LES COMMANDES CI-DESSOUS "F D",
  "X E", "E D" COMPORTENT UN CTRL-A
  EN PREMIER CARACTERE, JUSTE APRES
  LE PREMIER "
10 HIMEM: 4095:AD = 4096
11 PRINT CHR$(4)"PR#2": PRINT "F D"
12 PRINT "X E": PRINT "E D": PRINT CHR$(4)"PR#0"
20 PRINT CHR$(4)"IN#2"
30 CALL 768:Z = PEEK(6): IF Z = 127 THEN 50
40 POKE AD + 1,Z: PRINT CHR$(Z): I = I + 1: GOTO 30
50 PRINT : PRINT CHR$(4)"IN#0": PRINT : PRINT : INPUT "FICHER : ";FIC$ : INPUT "DRIVE : ";D

```

```

60 PRINT : INPUT "OK ? (O/...) ";Z$: IF Z$ < "O" THEN 50
100 PRINT : PRINT CHR$(4)"OPEN"FIC$, D"D: PRINT CHR$(4)"WRITE"FIC$: FOR J = 0 TO I - 1
105 PRINT CHR$( PEEK(AD + J));
107 NEXT : PRINT : PRINT CHR$(4)"CLOSE"
110 PRINT CHR$(4)"PR#0"

```

TEST

```

0300- 20 05 C2 29 7F 85 06 60
0300- 20 05 C2 JSR $C205
0303- 29 7F AND $7F
0305- 85 06 STA $06
0307- 60 RTS

```

PROGRAMME MAC

```

10 CLS:DEFINT A-Z:CALL MOVETO(100,100):CALL TEXTFONT(0):WIDTH "LPT1:".255:INPUT "Fichier ? "F$:OPEN "I",F$
20 IF EOF(1) THEN 230
30 LINE INPUT#1:E$
40 L=LEN(E$):IF L=0 THEN LPRINT:GOTO 20
50 FOR I=1 TO L:FOR D=0 TO 40:NEXT: A=ASC(MID$(E$,I,1)):IF A=32 THEN 220
60 IF A<127 THEN LPRINT CHR$(A):GOTO 220
70 IF A=143 THEN LPRINT CHR$(125):GOTO 220

```

```

80 IF A=142 THEN LPRINT CHR$(123):GOTO 220
90 IF A=136 THEN LPRINT CHR$(64):GOTO 220
100 IF A=157 THEN LPRINT CHR$(124):GOTO 220
110 IF A=141 THEN LPRINT CHR$(92):GOTO 220
120 IF A=137 THEN LPRINT CHR$(94):CHR$(97):GOTO 220
130 IF A=144 THEN LPRINT CHR$(94):CHR$(101):GOTO 220
140 IF A=148 THEN LPRINT CHR$(94):CHR$(105):GOTO 220
150 IF A=153 THEN LPRINT CHR$(94):

```

```

CHR$(111):GOTO 220
160 IF A=158 THEN LPRINT CHR$(94):CHR$(117):GOTO 220
170 IF A=138 THEN LPRINT CHR$(126):CHR$(97):GOTO 220
180 IF A=145 THEN LPRINT CHR$(126):CHR$(101):GOTO 220
190 IF A=149 THEN LPRINT CHR$(126):CHR$(105):GOTO 220
200 IF A=154 THEN LPRINT CHR$(126):CHR$(111):GOTO 220
210 IF A=159 THEN LPRINT CHR$(126):CHR$(117):
220 NEXT:LPRINT:GOTO 20
230 CLOSE#1:LPRINT "":CHR$(127):END

```


Les tokens du Basic Microsoft

Jean-Luc Bazanegue

Le terme "token" n'est certainement pas inconnu des lecteurs de Pom's. Cependant, pour ceux d'entre-vous qui nous rejoignent à l'occasion de ce numéro, il faut rappeler qu'un token est la représentation interne d'un mot-clé du Basic. Ainsi, en Basic Microsoft (et sur le Macintosh), l'instruction **PRINT** devient, lorsque vous appuyez sur la touche **RETURN** pour valider une ligne, un octet contenant la valeur 172 (\$AC en hexadécimal). Les tokens permettent un gain de place en mémoire non

négligeable et facilitent le travail de l'interpréteur.

Voici ce qui nous a poussé à rechercher les tokens du Basic Microsoft : nous disposions d'un Mac depuis quelques temps, mais cet ordinateur sans langage de programmation ressemblait étrangement à un presse-purée sans pommes de terre : nous l'avions donc quelque peu délaissé. Et puis, un matin, nous vîmes arriver une disquette contenant le Basic. Voilà qui était de nature à redonner

un peu de sang à la bête (au Mac bien sûr...). Malheureusement, la disquette en question nous est parvenue sans documentation : pour connaître la liste des mots-clé de ce langage, il fallait trouver la table des tokens.

Nous remercions ceux qui se sont jetés sur leurs photocopieuses afin de nous envoyer une copie de leur documentation, mais cela est inutile : depuis que nous avons établie la liste jointe, Microsoft nous a fait parvenir les précieux documents.

ABS	\$80	128	EXP	\$90	144	MERGE	\$F8 \$9E	248	158	SAVE	\$F8 \$AB	248	171		
ALL	\$F9 \$FF	249	253	FIELD	\$91	145	MID\$	\$A5	165	SETCURSOR	\$FB \$FD	251	253		
AND	\$F1	241	FILES	\$F8 \$94	248	148	MKD\$	\$A6	166	SGN	\$B4	180			
APPEND	\$F9 \$FE	249	254	FILLARC	\$FB \$D8	251	216	MK1\$	\$A7	167	SHOWCURSOR	\$FB \$FB	251	251	
AS	\$F9 \$FD	249	253	FILLOVAL	\$FB \$E2	251	226	MKS\$	\$A8	168	SHOWPEN	\$FB \$F8	251	248	
ASC	\$81	129	FILLRECT	\$FB \$E7	251	231	MOD	\$F6	246	SIN	\$B5	181			
ATN	\$82	130	FILLROUNDRECT	\$FB \$DD	251	221	MOUSE	\$F8 \$B8	248	184	SPACE\$	\$B6	182		
AUTO	\$F8 \$80	248	128	FIX	\$92	146	MOVE	\$FB \$F1	251	241	SPC	\$F9 \$FA	249	250	
BACKPAT	\$FB \$FF	251	255	FN	\$93	147	MOVETO	\$FB \$F2	251	242	SQR	\$B7	183		
BASE	\$F9 \$FC	249	252	FOR	\$94	148	NAME	\$F8 \$9F	248	159	STEP	\$F9 \$F9	249	249	
BEEP	\$F8 \$B5	248	181	FRAMEARC	\$FB \$DC	251	220	NEW	\$FB \$A0	248	160	STOP	\$F8 \$AD	248	173
CALL	\$83	131	FRAMEOVAL	\$FB \$E6	251	230	NEXT	\$A9	169	STR\$	\$B8	184			
CDL	\$84	132	FRAMERECT	\$FB \$EB	251	235	NOT	\$E7	231	STRING\$	\$B9	185			
CHAIN	\$F8 \$81	248	129	FRAMEROUNDRECT	\$FB \$E1	251	225	OBSCURECURSOR	\$FB \$FA	251	250	SWAP	\$F8 \$AE	248	174
CHK\$	\$85	133	FRE	\$F8 \$95	248	149	OCT\$	\$FB \$A1	248	161	SYSTEM	\$F8 \$AF	248	175	
CINT	\$86	134	GET	\$95	149	ON	\$AA	170	TAB	\$F9 \$F8	249	248			
CIRCLE	\$F8 \$B6	248	182	GETPEN	\$FB \$F7	251	247	OPEN	\$AB	171	TAN	\$BA	186		
CLEAR	\$F8 \$82	248	130	GOSUB	\$96	150	OPTION	\$F8 \$A2	248	162	TEXTFACE	\$FB \$EE	251	238	
CLOSE	\$87	135	GOTO	\$97	151	OR	\$F2	242	TEXTFONT	\$FD \$EF	251	239			
CLS	\$F8 \$83	248	131	HEX\$	\$F8 \$96	248	150	OUTPUT	\$F9 \$FB	249	251	TEXTMODE	\$FB \$ED	251	237
COMMON	\$88	136	HIDECURSOR	\$FB \$FC	251	252	PAINTARC	\$FB \$DB	251	219	TEXTSIZE	\$FB \$EC	251	236	
CONT	\$F8 \$84	248	132	HIDEPEN	\$FB \$F9	251	249	PAINTOVAL	\$FB \$E5	251	229	THEN	\$E6	230	
COS	\$89	137	IF	\$98	152	PAINTRECT	\$FB \$EA	251	234	TIME	\$F8 \$B0	248	176		
CSNG	\$F8 \$85	248	133	IMP	\$F5	245	PAINTROUNDRECT	\$FB \$E0	251	224	TIMER	\$F8 \$B1	248	177	
CVD	\$8A	138	INITCURSOR	\$FB \$FE	251	254	PEEK	\$F8 \$A3	248	163	TO	\$E5	229		
CVI	\$8B	139	INKEY\$	\$99	153	PENMODE	\$FB \$F5	251	245	TROFF	\$F8 \$B2	248	178		
CVS	\$8C	140	INPUT	\$9A	154	PENNORMAL	\$FB \$F3	251	243	TRON	\$F8 \$B1	248	177		
DATA	\$8D	141	INSTR	\$F8 \$97	248	151	PENPAT	\$FB \$F4	251	244	USING	\$E4	228		
DATE\$	\$F8 \$86	248	134	INT	\$9B	155	PENSIZE	\$FB \$F6	251	246	USR	\$F9 \$F7	249	247	
DEF	\$F8 \$8B	248	139	INVERTARC	\$FB \$D9	251	217	POINT	\$F8 \$B9	248	185	VAL	\$BC	188	
DEFDBL	\$F8 \$89	248	137	INVERTOVAL	\$FB \$E3	251	227	POKE	\$F8 \$A4	248	164	VARPTR	\$F8 \$B3	248	179
DEFINT	\$F8 \$87	248	135	INVERTRECT	\$FB \$E8	251	232	POS	\$F8 \$A5	248	165	WAIT	\$F9 \$F6	249	246
DEFSNG	\$F8 \$8A	248	136	INVERTROUNDRECT	\$FB \$DE	251	222	PRESET	\$F8 \$BA	248	186	WEND	\$BD	189	
DEFSTR	\$F8 \$8A	248	136	KILL	\$F8 \$98	248	152	PRINT	\$AC	172	WHILE	\$BE	190		
DELETE	\$F8 \$8C	248	140	LCOPY	\$F8 \$B7	248	183	PSET	\$F8 \$BB	248	187	WIDTH	\$F8 \$B4	248	180
DIM	\$F8 \$8D	248	141	LEFT\$	\$9C	156	PTAB	\$FB \$D7	251	215	WRITE	\$BF	191		
EDIT	\$F8 \$8E	248	142	LEN	\$9D	157	PUT	\$AD	173	XOR	\$F3	243			
ELSE	\$8E	142	LET	\$9E	158	RANDOMIZE	\$F8 \$A6	248	166	+ (plus)	\$EC	236			
END	\$F8 \$8F	248	143	LINE	\$9F	159	READ	\$AE	174	- (moins)	\$ED	237			
EOF	\$8F	143	LINETO	\$FB \$F0	251	240	REM	\$AF	175	* (astérisque)	\$EE	238			
EQV	\$F4	244	LIST	\$F8 \$99	248	153	RENUM	\$F8 \$A7	248	167	/	\$EF	239		
ERASE	\$F8 \$90	248	144	LLIST	\$F8 \$9A	248	154	RESET	\$F8 \$BC	248	188	' (puissance)	\$F0	240	
ERASEARC	\$FB \$DA	251	218	LOAD	\$F8 \$9B	248	155	RESTORE	\$F8 \$A8	248	168	\	\$F7	247	
ERASEOVAL	\$FB \$E4	251	228	LOC	\$A1	161	RESUME	\$F8 \$A9	248	169	(apostrophe)	\$E8	232		
ERASERECT	\$FB \$E9	251	233	LOF	\$A2	162	RETURN	\$B0	176	>	\$E9	233			
ERASEROUNDRECT	\$FB \$DF	251	223	LOG	\$A3	163	RIGHT\$	\$B1	177	<	\$EA	234			
ERL	\$F8 \$91	248	145	LPOS	\$F8 \$9C	248	156	RND	\$B2	178		\$EB	235		
ERR	\$F8 \$93	248	147	LPRINT	\$F8 \$9D	248	157	RSET	\$B3	179					
ERROR	\$F8 \$92	248	146	LSET	\$A4	164	RUN	\$FB \$AA	248	170					

Gestion de masques améliorée

Jacques Tran-Van
Maurice Schapiro

Pom's déclare que si, au cours d'une année, un seul des programmes présentés répond aux besoins du lecteur, il justifie à lui seul l'abonnement à la revue. Dans le Pom's 7, Gérard Michel nous a proposé un programme pour créer et modifier rapidement, à partir de masques, des écrans texte. Pour ceux qui, comme moi, accordent une importance primordiale à l'ergonomie des programmes, cet utilitaire tombe dans la catégorie des programmes qui, à eux seuls, ...

Pour tout programme dont on fait une utilisation intensive, il arrive vite un moment où l'on éprouve les limites du logiciel et où l'on souhaite les dépasser. Savoir ce qu'il faut modifier est une chose, le réaliser en est une autre. Aussi est-ce Jacques Tran-Van qui, par sa parfaite maîtrise de l'assembleur, a pu effectuer les modifications aboutissant à ce GESMASK MODIFIÉ que nous vous proposons.

Mode d'utilisation

Pour l'utiliser, il faut mettre sur la même disquette les programmes GESMASK MODIFIÉ, INPUT, HCT.OBJ, MOVESCREEN, GESMASK 1, GESMASK 2 et BLOAD-TEXT\$9200, ce dernier programme permettant de supprimer le bruit des drives au chargement des écrans.

L'ensemble fonctionne parfaitement bien avec PLE (voir Pom's 1) si vous prenez soin, durant la phase de conception de votre programme, de positionner HIMEM à 35840. Exemple :

```
96 REM Dans votre version définitive, HIMEM: 37376
97 HIMEM: 35840: PRINT CHR$(4) "BLOAD BLOAD-TEXT$9200"
98 POKE 782,136: CALL 768
99 REM Remettre M$=9200 dans la version définitive
100 D$=CHR$(4) "M$="A$8C00"
```

De cette façon, si au cours du déroulement de votre programme vous avez à nouveau besoin du dernier masque BLOADé, il sera inutile de demander à votre programme de le recharger; il suffira d'un simple CALL 768 pour le rappeler à l'écran.

Nouvelles commandes

Les commandes de GESMASK MODIFIÉ ci-dessous s'obtiennent en

```
1 *des des caract)res condens(s - marge pour l'imprimante
2
3
4 *****
5 *
6 *                               MASK POMS 7
7 *
8 *                               amélioré par TRAN-VAN Jacques
9 *                               et SCHAPIRO maurice
10 *                               le 15/12/83
11 *****
12
13
14 ORG $9200 ;Primitivement $927C
15
16
17 STOPFLG = $6 ;Drapeau d'arrêt du curseur
18 CARTYPE = $7 ;Drapeau INVERSE/NORMAL/FLASH
19 DEB = $8 ;Position de départ du décalage
20 SAVCAR = $9 ;Tampon de caract)re
21 SAVFLAG = $9 ;Drapeau > 0 range, sinon (change
22 SENSFLG = $9 ;Drapeau > 0 ins)re, < 0 efface
23 CVSAU = $19 ;Sauvegarde de la position vert.
24 PTR = $FE ;Pointeur de déplacement
25 CH = $24 ;Position hor. curseur
26 CV = $25 ;" vert.
27 BASL = $28 ;Adresse 1)re colonne
28 BASH = $29
29 BUFFER = $8000 ;Tampon d'écran - $400
30 BASCALC = $FBC1 ;Calcul de l'adresse de ligne
31 BS = $FC10 ;Envoi d'une <-
32 UTAB = $FC22 ;Déplace curseur en CV
33 HOME = $FC58
34 CLREOL = $FC9C ;Efface fin de ligne(A,Y d(é.)
35 RDKEY = $FD0C ;Entrée de caract)re
36 COUT = $FDED ;Sortie d'un caract)re
37
38
39 * Entrée principale du programme avec effacement
40 * de l'écran
41
42 INIT JSR HOME
43
44 * Autre entrée du programme sans effacement de l'écran
45 * (pour modification)
46
47 LDA #0
48 STA STOPFLG ;Initialise le drapeau
49 STA CARTYPE
50 DEBUT LDA CV
51 CMP #24 ;Dernière ligne ?
52 BNE TESTPOS
53 LDA #0 ;Retour 1)re col./1)re ligne
54 STA CV
55 BEQ GETCMD
56
57 TESTPOS LDA CH
58 CMP #28 ;Dernière col. de la ligne ?
59 BNE GETCMD
60
61 * Passe à la ligne suivante
62
63 NEXTLIN INC CV
64 LDA #0 ;Passer à 1)re col. ligne suivante
65 STA CH
66 BEQ DEBUT
67
68 * Lecture de la commande
69 * Autre entrée du programme(pour consultation)
70
71 GETCMD JSR UTAB ;Positionne le curseur
72 JSR RDKEY ;Entrée caract)re
73 CMP #""
74 BNE RETURN
75 LDA #$DB ;Crochet fermant
76
77 RETURN CMP #$80 ;Return ?
78 BEQ NEXTLIN ;A la ligne !
79
80 * Sortie du programme
81
82 EXITPGM CMP #$80 ;CTRL-SHIFT-P?
83 BNE NOMOVE
84 RTS
85
86 * Immobilisation du curseur
87
88 NOMOVE CMP #$93 ;CTRL-S ?
89 BNE DOWN
90 STA STOPFLG ; < 0
91 BEQ GETCMD
92
93 * Curseur une ligne plus bas
94
```

mode CTRL : on maintient la touche CTRL enfoncée, tout en appuyant sur la lettre de la commande.

I : inverse
F : flash
N : normal
W : monter
Z : descendre
D : décalage à droite
G : décalage à gauche
V : répétition verticale
Y : insertion dans la ligne
C : sort du mode insertion
O : insertion d'une ligne vide
P : suppression d'une ligne
R : sauvegarde du masque dans un buffer
X : rappel du masque placé dans le buffer
E : efface un caractère à partir du curseur
K : efface la ligne à partir du curseur
S : empêche le curseur de bouger après la frappe d'un caractère. On sort de ce mode avec les flèches gauche et droite

La touche RETURN renvoie le curseur au début de la ligne suivante. SHIFT-M engendre un crochet ouvrant, et SHIFT-N un crochet fermant.

Sauvegarde du masque

CTRL-SHIFT-P pose la question du nom à donner au masque. Quatre réponses sont possibles :

- RETURN : le masque est enregistré sur la disquette sous le nom que l'on a donné dans le menu principal;
- un nom : ce nom est alors attribué au masque sauvegardé sur la disquette. On peut ainsi mémoriser des variantes du masque appelé dans le menu principal;
- # (dièse) : annule le masque en cours et revient au menu principal (le masque n'est alors pas sauvegardé);
- ? : affiche le catalogue de la disquette puis repose la question.

A titre de démonstration, vous trouverez le petit programme en Basic FRANCE, qui apprendra à vos enfants à situer les départements sur la carte de France. Pour fonctionner, il requiert le masque CARTE. L'utilisation de ce programme est simple et ne nécessite aucun commentaire.

Routine INPUT

*9000.9089

```
9000- A9 80 A6 07 9D 11 03 CA
9008- 10 FA A2 00 86 06 20 0C
9010- FD C9 8D F0 72 E4 07 90
9018- 0A C9 88 F0 28 20 DD FB
9020- 4C 0E 90 C9 9B D0 09 E0
9028- 00 D0 05 A9 09 85 06 60
```

```
95 DOWN      CMP    #$9A      ;CTRL-Z ?
96           BNE    FLECHDR
97           INC    CV          ;Ligne suivante
98           BNE    DEBUT
99
100 * Avance le curseur d'un caractère
101
102 FLECHDR    CMP    #$95      ;'-' ?
103           BNE    BACK
104           LSR    STOPFLG     ;Avance normale du curseur
105           INC    CH          ;Caractère suivant
106 EXFLECH    BPL    TESTPOS
107
108 * Recule le curseur d'un caractère
109
110 BACK       CMP    #$88      ;'<' ?
111           BNE    UP
112           JSR    BS          ;Recul du curseur
113           LSR    STOPFLG     ;Avance normale du curseur
114           BPL    GETCMD
115
116 * Curseur une ligne plus haut
117
118 UP         CMP    #$97      ;CTRL-W ?
119           BNE    INVERSE
120           DEC    CV          ;Ligne précédente
121           BPL    GETCMD      ;Si CV=0 on positionne en CV
122           LDA    #23        ;On saute à la dernière ligne
123           STA    CV
124           GETCMD
125
126 * Ecriture en inverse
127
128 INVERSE    CMP    #$89      ;CTRL-I ?
129           BNE    FLASH
130           LDA    #1
131 MODE       STA    CARTYPE     ;Flag INVERSE
132           JMP    GETCMD
133
134 * Ecriture en flash
135
136 FLASH      CMP    #$86      ;CTRL-F
137           BEQ    MODE         ;Flag < 0 FLASH
138
139 * Ecriture en normal
140
141 NORMAL     CMP    #$8E      ;CTRL-N ?
142           BNE    DUPLIC
143           LDA    #0          ;Flag NORMAL
144           BEQ    MODE
145
146 * Duplication du caractère une ligne plus bas
147
148 DUPLIC     CMP    #$96      ;CTRL-V ?
149           BNE    EOL
150           LDA    CV
151           STA    STOPFLG     ;Avance normale du curseur
152           CMP    #23        ;Dernière ligne => on ignore
153 RELAI      BCS    GETCMD
154           JSR    BASCALC     ;Calcul adresse base ligne CV
155           LDY    CH
156           LDA    (BASL),Y    ;Charge le car. sous le curseur
157           INC    CV          ;Ligne suivante
158           JSR    OUTPUT      ;Affiche le caractère
159           DEC    CH          ;Recul du curseur
160 NOINC      BPL    EXFLECH    ;Retour en TESTPOS
161
162 * Effacement de la fin de la ligne
163
164 EOL        CMP    #$86      ;CTRL-K ?
165           BNE    RIGHT
166           JSR    CLREOL      ;Efface la fin de la ligne
167           BCS    RELAI       ;Retour en GETCMD
168
169 * Déplacement de la fin de l'(cran d'un caractère)
170 * vers la droite
171
172 RIGHT      CMP    #$84      ;CTRL-D ?
173           BNE    LEFT
174           LDA    CV
175           PHA
176           LDA    #" "       ;Sauve la position verticale
177           STA    SAVCAR
178           LDY    CH
179 DROITE     STY    DEB
180           JSR    UTAB
181           JSR    LIGNEDR     ;Décale la ligne à droite
182           LDY    #0
183           INC    CV          ;Ligne suivante
184           LDA    CV
185           CMP    #24        ;Dernière ligne
186           BCC    DROITE
187           PLA
188           STA    CV          ;Récupère la position verticale
189           BCS    RELAI       ;JMP GETCMD
190
191 * Déplacement de la fin de l'(cran d'un caractère)
192 * vers la gauche
193
194 LEFT       CMP    #$87      ;CTRL-G ?
195           BNE    INSERER
196           LDA    #" "
197           STA    SAVCAR
198           LDA    CV
199           STA    CVSAV
200           LDA    #23
201           STA    CV
202 GAUCHE     LDY    CH
203           LDA    CV
204           CMP    CVSAV
205           BEQ    LAST
```

```

206      BCC  EXLEFT
207      LDY  #0
208      LAST JSR  UTAB
209      JSR  LIGNEGA ;D(calle la ligne 2 gauche
210      DEC  CV
211      BPL  GAUCHE
212      EXLEFT LDA  CVSAV
213      STA  CV
214      EXGAUCH JMP  GETCMD
215
216      * Insertion d'un caractere dans une ligne
217
218      INSERER CMP  #$99 ;CTRL-Y
219      BNE  DELETE
220      LOOPINS JSR  RDKEY
221      CMP  #$83 ;CTRL-C ?
222      BEQ  EXGAUCH
223      LDY  CH
224      STY  DEB
225      CPY  #39
226      BCC  NOLAST
227      STA  STOPFLG ;On n'avance pas si dernier car.
228      JSR  OUT
229      LSR  STOPFLG
230      BPL  LOOPINS
231      NOLAST PHA
232      JSR  LIGNEDR ;D(calle la ligne vers la droite
233      PLA
234      JSR  OUT
235      BPL  LOOPINS ;Boucle
236
237      * Effacement d'un caractere de la ligne
238      * et d(callage de la fin vers la gauche
239
240      DELETE CMP  #$85 ;CTRL-E ?
241      BNE  ECHANGE
242      LDA  # " "
243      STA  SAVCAR
244      LDY  CH
245      JSR  LIGNEGA
246      BCC  BAK
247
248      * Echange de l'(cran actuel et de celui
249      * sauve dans le tampon
250
251      ECHANGE CMP  #$98 ;CTRL-X ?
252      BNE  RANGER
253      STA  SAVFLAG ;Flag < 0
254      DOIT LDX  #23
255
256      NEXTLG TXA
257      JSR  BASCALC
258      LDA  BASL
259      STA  PTR
260      CLC
261      LDA  BASH
262      ADC  #>BUFFER
263      STA  PTR+1
264      LDY  #39
265      LOOPCHG LDA  (BASL),Y
266      BIT  SAVFLAG
267      BPL  NOTCHANG
268      PHA
269      LDA  (PTR),Y
270      STA  (BASL),Y
271      PLA
272      NOTCHANG STA  (PTR),Y
273      DEY
274      BPL  LOOPCHG
275      DEX
276      BPL  NEXTLG
277      BMI  BAK
278
279      * Sauvegarde l'(cran actuel dans un tampon
280
281      RANGER CMP  #$92 ;CTRL-R ?
282      BNE  LININSRT
283      LSR  SAVFLAG
284      BPL  DOIT ;Se branche toujours
285
286      * Insertion d'une ligne blanche et d(callage
287      * de la fin de l'(cran d'une ligne vers le bas
288
289      LININSRT CMP  #$8F ;CTRL-O ?
290      BNE  LINEKILL
291      LDX  #22
292      STX  SENSFLG ;Flag > 0 => vers le bas
293      DPL JSR  MOVELINE ;D(place la ligne
294      DEX
295      DEX
296      BMI  BLANKLIN
297      CPX  CV
298      BCS  DPL
299      BLANKLIN JSR  BLANK ;Efface la ligne
300      BMI  BAK
301
302      * Effacement d'une ligne et d(callage de la fin
303      * de l'(cran d'une ligne vers le haut
304
305      LINEKILL CMP  #$90 ;CTRL-P ?
306      BNE  DSPCAR
307      STA  SENSFLG ;Flag < 0 => vers le haut
308      LDX  CV
309      NXTPRO INX
310      CPX  #24
311      BNE  KILLNXT
312      LDA  #23
313      JSR  BLANK1 ;Efface la derniere ligne
314      BMI  BLANKLIN ;Toujours
315      KILLNXT JSR  MOVELINE ;D(place la ligne
316      INX

```

```

9030- C9 AC D0 05 A9 AE 4C 7D
9038- 90 C9 AE F0 40 C9 88 D0
9040- 0A E0 00 F0 04 CA C6 24
9048- 4C 0E 90 C9 95 D0 0C A5
9050- 25 20 C1 FB A4 24 B1 28
9058- 4C 7D 90 C9 A0 90 AF C9
9060- DB 80 AB A4 08 C0 01 F0
9068- 0C C9 AF 80 04 C9 AD D0
9070- 7D C9 BA 80 99 E0 00 D0
9078- 04 C9 A2 F0 91 20 ED FD
9080- 9D 11 03 E8 4C 0E 90 86
9088- 09 60

```

Routine MOVESCREEN

*2DC.2FD

```

02DC- A9 00 85 FC
02E0- 85 FE A8 AD FE 02 85 FF
02E8- AD FF 02 85 FD A2 04 B1
02F0- FE 91 FC C8 D0 F9 E6 FD
02F8- E6 FF CA D0 F2 60

```

Routine HCT.OBJ

*9465.9496

```

9465- A9 00 85
9468- 25 A0 00 20 C1 FB B1 28
9470- C9 40 80 0D C9 20 80 06
9478- 18 69 C0 4C B1 94 18 69
9480- 80 20 ED FD C8 C0 28 D0
9488- E5 A9 8D 20 ED FD E6 25
9490- A5 25 C9 18 D0 D3 60

```

Routine BLOADTEXT

\$9200

*300.31D

```

0300- A2 17 8A 20 C1 FB A5 28
0308- 85 FE 18 A5 29 69 8E 85
0310- FF A0 27 B1 FE 91 28 88
0318- 10 F9 CA 10 E5 60

```

Récapitulation MASK

*9200.941E

```

9200- 20 58 FC A9 00 85 06 85
9208- 07 A5 25 C9 18 D0 04 A9
9210- 00 85 25 F0 0E A5 24 C9
9218- 28 D0 08 E6 25 A9 00 85
9220- 24 F0 E6 20 22 FC 20 0C
9228- FD C9 DE D0 02 A9 DB C9
9230- 8D F0 E8 C9 80 D0 01 60
9238- C9 93 D0 04 85 06 F0 E3
9240- C9 9A D0 04 E6 25 D0 C1
9248- C9 95 D0 06 46 06 E6 C4
9250- 10 C3 C9 88 D0 07 20 10
9258- FC 46 06 10 C6 C9 97 D0
9260- 0A C6 25 10 BE A9 17 85
9268- 25 D0 88 C9 89 D0 07 A9
9270- 01 85 07 4C 23 92 C9 86
9278- F0 F7 C9 8E D0 04 A9 00
9280- F0 EF C9 96 D0 18 A5 25
9288- 85 06 C9 17 B0 95 20 C1
9290- FB A4 24 B1 28 E6 25 20
9298- A4 93 C6 24 10 B2 C9 88
92A0- D0 05 20 9C FC B0 E5 C9
92A8- 84 D0 20 A5 25 48 A9 A0

```



```

317          BPL NXTPRO      ;Toujours
318
319 * Affichage du caractère
320
321 DSPCAR    JSR    OUT
322 BAK       JMP    TESTPOS
323
324 OUT       LDX    CARTYPE
325
326 * CARTYPE=1 => inverse CARTYPE=0 => normal
327 * CARTYPE < 0 => flash
328
329          BEQ    OUTPUT      ;Normal
330          BMI    CLIGNO
331          AND    #$3F
332          ;Inverse
333
334 * Affichage du caractère et avance curseur
335
336 OUTPUT    PHA
337          LDA    CU          ;Sauve le caractère
338          JSR    BASCALC
339          LDY    CH
340          PLA          ;Récupère le caractère
341          STA    (BASL),Y
342          BIT    STOPFLG
343          BMI    RETOUR
344          INC    CH
345
346 RETOUR    RTS
347
348 CLIGNO    AND    #$3F      ;En flash
349          ORA    #$40
350          BNE    OUTPUT
351
352 * Décalage de la ligne d'un caractère vers la droite
353
354 LIGNEDR   LDY    #39
355
356 PUSHDR    LDA    (BASL),Y   ;Sauve les caractères sur la pile
357          PHA
358          DEY
359          BMI    DR
360          CPY    DEB
361          BCS    PUSHDR
362          LDA    SAVCAR
363          INY
364          STA    (BASL),Y   ;Décale vers la droite
365          PLA
366          INY
367          CPY    #40
368          BCC    PULLDR
369          STA    SAVCAR
370          RTS
371
372 * Décalage de la ligne d'un rang vers la gauche
373
374 LIGNEGA   STY    DEB
375
376 PUSHGA    LDA    (BASL),Y
377          PHA
378          INY
379          CPY    #40
380          BCC    PUSHGA
381          LDA    SAVCAR
382          DEY
383          STA    (BASL),Y
384          PLA
385          BMI    STOPJLL
386          CPY    DEB
387          BCS    PULLGA
388          STA    SAVCAR
389          CLC
390          RTS
391
392 * Déplacement d'une ligne vers le haut
393 * ou vers le bas
394
395 MOVELINE TXA          ;No. de la ligne
396          JSR    BASCALC

```

```

9280- 85 09 A4 24 84 08 20 22
9288- FC 20 BC 93 A0 00 E6 25
92C0- A5 25 C9 18 90 EE 68 85
92C8- 25 80 C1 C9 87 D0 29 A9
92D0- A0 85 09 A5 25 85 19 A9
92D8- 17 85 25 A4 24 A5 25 C5
92E0- 19 F0 04 90 0C A0 00 20
92E8- 22 FC 20 D6 93 C6 25 10
92F0- EA A5 19 85 25 4C 23 92
92F8- C9 99 D0 22 20 0C F0 C9
9300- 83 F0 F2 A4 24 84 08 C0
9308- 27 90 09 85 06 20 9C 93
9310- 46 06 10 E8 48 20 8C 93
9318- 68 20 9C 93 10 DE C9 85
9320- D0 0B A9 A0 85 09 A4 24
9328- 20 D6 93 70 6C C9 98 D0
9330- 28 85 09 A2 17 8A 20 C1
9338- FB A5 28 85 FE 18 A5 29
9340- 69 80 85 FF A0 27 B1 28
9348- 24 07 10 06 48 B1 FE 91
9350- 28 68 91 FE 88 10 EF CA
9358- 10 DB 30 3D C9 92 D0 04
9360- 46 09 10 CF C9 8F D0 14
9368- A2 16 86 09 20 F1 93 CA
9370- CA 30 04 E4 25 B0 F5 20
9378- 12 94 30 1D C9 90 D0 16
9380- 85 09 A6 25 E8 E0 18 D0
9388- 07 A9 17 20 14 94 30 E7
9390- 20 F1 93 E8 10 EE 20 9C
9398- 93 4C 15 92 A6 07 F0 04
93A0- 30 14 29 3F 48 A5 25 20
93A8- C1 FB A4 24 68 91 28 24
93B0- 06 30 02 E6 24 60 29 3F
93B8- 09 40 D0 E8 A0 27 B1 28
93C0- 48 88 30 04 C4 08 B0 F4
93C8- A5 09 C8 91 28 68 C8 C0
93D0- 28 90 F8 85 09 60 84 08
93D8- B1 28 48 C8 C0 28 90 F8
93E0- A5 09 88 91 28 68 88 30
93E8- 04 C4 08 B0 F6 85 09 18
93F0- 60 8A 20 C1 FB A0 27 B1
93F8- 28 48 88 10 FA 24 09 10
9400- 03 CA 10 01 E8 8A 20 C1
9408- FB C0 68 91 28 C0 27 90
9410- F8 60 A5 25 20 C1 FB A9
9418- A0 91 28 88 10 FB 60

```

```

397          LDY    #39
398          LDA    (BASL),Y
399          PHA
400          DEY
401          BPL    LOOP1
402          BIT    SENSFLG
403          BPL    INSRT
404          DEX
405          BPL    CONTMOV
406          INSRT INX
407          CONTMOV TXA
408          JSR    BASCALC
409          LOOP2 INY
410          PLA
411          STA    (BASL),Y
412          CPY    #39
413          BCC    LOOP2
414          RTS
415
416 * Effacement de la ligne
417
418 BLANK     LDA    CU
419 BLANK1    JSR    BASCALC
420          LDA    #""
421          BKLOOP STA    (BASL),Y
422          DEY
423          BPL    BKLOOP
424          RTS
425
426          LST    OFF

```

GESMASK MODIFIE

```

10 HIMEM: 32768: REM    $8000
15 MOVE = 732:DP = 766:BT = 767
20 D$ = CHR$(4):D1$ = CHR$(13) + D$:
    PRINT D$"BLOAD INPUT": PRINT D$
    BLOAD MASK": PRINT D$"BLOAD HCT.0
    BJ": PRINT D1$"BLOAD MOVESCREEN"
30 ONERR GOTO 9000
40 GOTO 300
70 VTAB 22: HTAB 1: INVERSE : PRINT ZM$
    ;: NORMAL : INPUT " ? ";Z$: VTAB
    22: CALL - 868:Z$ = LEFT$(Z$,1
    ): IF Z$ = "0" OR Z$ = "N" THEN
        RETURN
71 GOTO 70

```

```

80 REM
90 REM ROUTINE D'INPUT
100 REM
149 VTAB V: HTAB H: PRINT LEFT$(PO$,L
    0)" "; VTAB V: HTAB H: IF ME =
    2 OR X = 1 THEN PRINT ZY$(LL): I
    F B% = 8 THEN 168
150 VTAB V: POKE 8,TY: POKE 36,H - 1: P
    OKE 7,LO: CALL 36864
154 LC = PEEK(9):E = PEEK(6): IF E =
    9 THEN RETURN
156 Z2$ = "": PRINT SPC(LO - LC): IF L
    C = 0 THEN 168
158 FOR Z2 = 1 TO LC:Z = PEEK(784 + Z
    2) - 128:Z2$ = Z2$ + CHR$(Z): N
    EXT

```

LIST

LIST

LE JOURNAL
DES AMATEURS
DE PROGRAMMATION

n°1

le journal des amateurs de programmation

Si programmer
un ordinateur est
devenu pour vous
un loisir, un plaisir...

une passion, sachez que **LIST** a été créé
pour vous. **LIST** vous aide à tirer davantage
de votre matériel, à vous perfectionner
dans la conception des programmes
qui "tourneront" sur votre machine.
LIST vous initie aux langages informatiques
et sélectionne les meilleurs livres pour
progresser. **LIST** vous informe de l'actualité
et vous fournit trucs, astuces et idées
pour mieux programmer...
Pour être sûr de ne rater aucun numéro
et pour recevoir **LIST** chez vous,
abonnez-vous !

**LIST, LE PLAISIR DE
PROGRAMMER**
20F chez votre marchand de journaux



**FAITES
40F
D'ECONOMIE!**

BULLETIN D'ABONNEMENT

à retourner à **LIST**

(service abonnement)

5, place du Colonel-Fabien,
75491 Paris Cedex 10

Nom : _____

Adresse : _____

Ville : _____

Code postal : _____ Pays : _____

Veillez m'abonner pour 10 numéros au prix
avantageux de **160 F*** au lieu de 200 F. Je fais ainsi
une économie de 40 F sur le prix de vente au numéro.
Je joins mon règlement indispensable libellé
à l'ordre de **LIST**.

* Belgique : 1330 FB ; Suisse : 50 FS ; Canada : 30 \$C ;
autres pays : 210 FF

```

162 IF TY < 3 THEN RETURN
163 IF MID$(ZZ$,2,1) = "/" THEN Z2$ =
    "0" + Z2$
164 IF MID$(ZZ$,5,1) = "/" THEN Z2$ =
    LEFT$(Z2$,3) + "0" + RIGHT$(
        Z2$,4)
165 IF LEN(Z2$) < > 8 THEN Z$ = ME$(
    3): GOSUB 190: GOTO 149
166 Z4 = VAL(MID$(Z2$,4,2)): IF VAL
    (LEFT$(Z2$,2)) > 31 OR Z4 > 12
    OR Z4 * VAL(RIGHT$(Z2$,2)) <
    = 0 THEN Z$ = ME$(3): GOSUB 190
    : GOTO 149
167 RETURN
168 IF ME = 2 OR X = 1 THEN VTAB V: HT
    AB H: Z2$ = ZY$(LL): PRINT Z2$ SPC
    (LO - LEN(Z2$))
169 RETURN
180 REM
190 VTAB 21: HTAB 1: CALL - 868: INVER
    SE: FOR Z = 1 TO 150: Z1 = PEEK
    (-16336): NEXT: PRINT Z$: NOR
    MAL: FOR Z = 1 TO 2500: NEXT: H
    TAB 1: CALL - 868: RETURN
300 DATA 9,28,1,2,17,36,1,1
310 FOR I = 1 TO 2: READ ZV%(I), ZH%(I),
    ZL%(I), TY%(I): NEXT
320 PO$ = "....."
400 TEXT: HOME: PRINT D1$ "BLOAD GESMA
    SK1,D1": REM PREMIER MASQUE DU
    PROGRAMME
405 V = 14: H = 28: LO = 1: TY = 2: X = 0: ME
    = 0: B% = 0
410 E = 0: GOSUB 149: IF E = 9 THEN 410
420 Z% = VAL(Z2$): IF Z% < 1 OR Z% > 6
    THEN 410
430 IF Z% = 6 THEN HOME: END
440 V = 17: H = 24: LO = 9: TY = 1: LL = 1
450 E = 0: GOSUB 149: IF E = 9 THEN 405
460 ZY$(1) = Z2$: M$ = Z2$: ZM$ = "ENREGIS
    TREMENT CONFIRME": GOSUB 70: IF Z
    $ = "N" THEN X = 1: GOTO 450
470 HOME: PRINT D$ "BLOAD GESMASK2": X =
    0: ME = 0: B% = 0: REM DEUXIEME M
    ASQUE DU PROGRAMME
480 V = 7: H = 29: TY = 2: LO = 1: E = 0: GO
    SUB 149: IF E = 9 THEN 400
490 ZY$(1) = Z2$: D = VAL(Z2$): IF D <
    1 OR D > 2 THEN X = 1: GOTO 480
500 IF Z% < > 4 THEN 560
510 ME = 0: D% = 2: X = 0
520 FOR LL = D% TO 3: L1 = LL - 1: V = ZV
    %(L1): H = ZH%(L1): LO = ZL%(L1): TY
    = TY%(L1)
525 E = 0: GOSUB 149: IF E = 9 AND LL =
    2 THEN LL = 3: NEXT: X = 1: GOTO
    480
530 IF E = 9 THEN LL = 1: X = 1: GOTO 55
    0
540 ZY$(LL) = Z2$
550 NEXT: S = VAL(ZY$(2)): E$ = ZY$(3)
    : IF S < 1 OR S > 5 OR (E$ < > "
    0" AND E$ < > "N") THEN D% = 3: X
    = 1: GOTO 520
560 GOSUB 70: IF Z$ = "N" THEN X = 1: G
    OTO 480
570 Z = 0: HOME: PRINT D1$ "BLOAD M$", D
    D: ON Z% GOTO 600,610,620,630,650
600 IF Z = 0 THEN 9010
602 CALL 37376: REM ROUTINE DE GESTIO
    N DE MASQUES

```

```

603 POKE DP,4: POKE BT,136: CALL MOVE:
    REM DEPLACEMENT DE $400 EN $8800
604 HOME: VTAB 10
605 INPUT "SOUS QUEL NOM LE SAUVER: ";
    NW$: IF NW$ = "?" THEN PRINT D1$
    "CATALOG": GET WW$: PRINT: GOTO
    605
606 IF NW$ = "H" THEN 400
607 IF NW$ < > "" THEN M$ = NW$
608 POKE DP,136: POKE BT,4: CALL MOVE
609 POKE 34,24: PRINT D$ "BSAVE M$", A102
    4, L1024: GOTO 400
610 CALL 37379: GOTO 603: REM ON SAUTE
    LE "HOME" DE LA ROUTINE
620 POKE 36,0: POKE 37,23: CALL 37411:
    GOTO 400: REM ONSAUTE DIRECTEMENT A
    U"JSR VTAB" DE LA ROUTINE
630 PRINT D$ "PR#": S: IF E$ = "0" THEN P
    RINT CHR$(9) "80N"
640 CALL 37989: PRINT D1$ "PR#0": GOTO 4
    00: REM ROUTINE DE HARD-COPY TEX
    T
650 ZM$ = "ANNULATION CONFIRMEE": GOSUB
    70: IF Z$ = "N" THEN 400
660 PRINT D1$ "DELETE M$": GOTO 400
9000 Z = PEEK(222): IF Z = 6 AND Z% =
    1 THEN 600
9010 IF Z% = 1 THEN Z$ = "CE MASQUE EXI
    STE DEJA": GOSUB 190: GOTO 400
9020 Z$ = "CE MASQUE N'EXISTE PAS": GOSU
    B 190: GOTO 400

```

```

*****
*                                     *
*          GESTION DE MASQUES          *
*                                     *
*-----*
*                                     *
* 1 - CREATION          2 - MODIFICATION *
*                                     *
* 3 - CONSULTATION     4 - IMPRESSION   *
*                                     *
* 5 - ANNULATION       6 - FIN          *
*                                     *
*          VOTRE CHOIX :              *
*                                     *
*          NOM DU MASQUE : .....      *
*                                     *
*****
*****
*          GESTION DE MASQUES - PARAMETRES *
*                                     *
*-----*
*          NUMERO DU DRIVE (1/2) : .    *
*                                     *
*          SLOT DE L'IMPRIMANTE : .     *
*                                     *
*-----*
*          VOTRE INTERFACE-IMPRIMANTE FAIT *
*                                     *
*          ELLE UN AFFICHAGE A L'ECRAN AU *
*                                     *
*          MOMENT DE L'IMPRESSION (O/N) : *
*                                     *
*****

```


Visicalc est vraiment formidable ! Pour la gestion financière ou commerciale bien sûr, mais également pour bien d'autres choses : Olivier Herz vous a présenté un calendrier perpétuel dans le numéro 10 de Pom's; aujourd'hui, je vous propose de calculer les heures de lever et coucher du soleil.

Pour utiliser ce tableau, il suffit de le charger à partir de la disquette Pom's; si vous ne la recevez pas, vous trouverez ci-dessous la liste du tableau obtenue avec le programme LISTE de la disquette d'accompagnement "Visicalc sur Apple" (Editions du PSI).

La feuille affiche deux fenêtres : sur la première, vous entrez le jour et le mois, la latitude et longitude du point considéré (format degrés.minutes, positif pour le nord et l'ouest, négatif pour le sud et l'est) ainsi que le décalage horaire officiel par rapport au méridien de Greenwich.

Le tableau utilise par défaut les coordonnées géographiques de Paris; les coordonnées de quelques grandes villes en France et ailleurs se trouvent sur la droite du tableau; vous pouvez bien entendu en ajouter d'autres en consultant un atlas. N'oubliez pas d'indiquer le décalage horaire (+1 en hiver et +2 en été pour la France); vous pourrez trouver celui de n'importe quel pays sur un indicateur d'Air France.

La deuxième fenêtre affiche les résultats en heures locales sous le format heures.minutes, ainsi que la durée totale du jour.

Lève-toi et brille !

Alexandre Avrane

Quelques tests sont inclus pour vérifier la saisie; le message "Error" s'affiche en cas d'incohérence. Dans certains cas extrêmes (en particulier en Sibérie), il est possible que le soleil se lève avant 0 heure ou se couche après minuit; la grille affichera alors un astérisque à la suite de l'heure locale.

Attention : l'aplatissement des pôles

et la réfraction atmosphérique ne sont pas pris en compte et peuvent conduire à des erreurs de l'ordre de 5 minutes. Enfin, cette grille ne tourne pas sur certains simili-Visicalc, en particulier Magicalc qui n'accepte pas les formules trigonométriques, ou Multiplan qui ignore Arcosinus; il faudrait revoir la grille et utiliser les développements limités : un lecteur relèvera-t-il le défi ? ■

	A	B	C	D
1	HEURES DE LEVER & COUCHER DU SOLEIL			
2	-----			
3	COPYRIGHT A. AVRANE ET POM'S - 1984			
4	=====			
5				
6	JOUR	----->	14	
7	MOIS	----->	7	
8	LATITUDE (DD.MM)->		48.52	N=+ S=-
9	LONGITUDE (DD.MM)->		-2.23	W=+ E=-
10	DECALAGE HORAIRE->		2.00	
11				
	A	B	C	D
16				
17	RESULTATS (HEURES LOCALES HH.MM)			
18	=====			
19				
20	LEVER	----->	6.09	
21	COUCHER	----->	21.44	
22	DUREE DU JOUR	--->	15.35	

FICHER VISICALC : SOLEIL

```
/X!/X>A16:>B19;;
/GC9
/X!/X>A1:>C6::/GFG
>B5:/TH
/X>A1:>B1:/TV
/GC9
/GFL
/XH12
/GRA
/GOR
/W1
>A1:"HEURES DE
>B1:" LEVER &
>C1:" COUCHER D
>D1:" U SOLEIL
>E1:"!
>A2:"-----
>B2:"-----
>C2:"-----
>D2:"-----
>E2:"!
>A3:"COPYRIGHT
```

```
>B3:" A. AVRANE
>C3:" E ET POM'
>D3:" S - 1984
>E3:"!
>A4:"=====
>B4:"=====
>C4:"=====
>D4:"=====
>E4:"!
>A6:"JOUR ----
>B6:"----->
>C6:/FL14
>D6:/F*QIF(@OR(C6<1,C6>31),@ERROR,0)
>F6:"PARIS
>G6:"MARSEILLE
>H6:"LILLE
>I6:"BREST
>J6:"STRASBOUR
>K6:"G OSLO
>L6:"ATHENES
>M6:"SINGAPOUR
>N6:"LE CAP
```

```

>O6:"MELBOURNE
>P6:" NEW YORK
>Q6:" ANCHORAG
>R6:"E LIMA
>S6:"TAHITI
>A7:"MOIS ----
>B7:"----->
>C7:/FL7
>D7:/F*@1F(@OR(C7<1,C7>12),@ERROR,0)
>A8:/FL"LATITUDE
>B8:"(DD.MM)->
>C8:/FS48.52
>D8:" N=+ S=-
>F8:48.52
>G8:43.26
>H8:50.34
>I8:48.27
>J8:48.32
>K8:59.54
>L8:37.54
>M8:1.21
>N8:-33.58
>O8:-37.44
>P8:40.46

```

```

>Q8:61.11
>R8:-12.02
>S8:-17.33
>A9:"LONGITUDE
>B9:"(DD.MM)->
>C9:/FS-2.23
>D9:" W=+ E=-
>F9:-2.23
>G9:-5.13
>H9:-3.05
>I9:4.25
>J9:-7.38
>K9:-10.38
>L9:-23.44
>M9:-103.54
>N9:-18.36
>O9:-144.54
>P9:73.53
>Q9:149.59
>R9:77.07
>S9:149.37
>A10:"DECALAGE
>B10:"HORAIRE->
>C10:/FS2

```

```

>D10:/F*@1F(@OR(C8<-90,C8>90,C9<-90,C9>90,C10<-12,C10>12),@ERROR,0)
>A12:/F**+(C8-@INT(C8))/6
>B12:/F**+(C9-@INT(C9))/6
>C12:/F**+@INT(C8)+A12
>D12:/F**+@INT(C9)+B12
>E12:/F*@PI/180
>A13:/F**+.988*(C6+(30.3*(C7-1)))
>B13:/F**+23.5*@COS((A13+10)*E12)
>C13:/F**+(.123*@COS((A13+87)*E12)-((1/6)*@SIN(2*(A13+10)*E12))
>D13:/F**+(180/@PI)*@ACOS(@TAN(B13*E12)*@TAN(C12*E12))
>A14:/F**+12+C10-C13+((1/15)*(D12-D13))
>B14:/F*@1F(A14<0,24+A14,A14)
>C14:/F**+12+C10-C13+((1/15)*(D12+D13))
>D14:/F*@1F(C14>24,C14-24,C14)
>A15:/F**+C14-A14
>A17:"RESULTATS
>B17:" (HEURES
>C17:"LOCALES H
>D17:"H.MM)
>A18:"=====
>B18:"=====
>C18:"=====
>D18:"=====
>A20:"LEVER ---
>B20:"----->
>C20:/FS+@INT(B14)+((6/10)*(B14-@INT(B14)))
>D20:/F*@1F(A14<0,@EXP(99),0)
>E20:"VEILLE
>A21:"COUCHER -
>B21:"----->
>C21:/FS+@INT(D14)+((6/10)*(D14-@INT(D14)))
>D21:/F*@1F(C14>24,@EXP(99),0)
>E21:"LENDEMAIN
>A22:"DUREE DU
>B22:"JOUR --->
>C22:/FS+@INT(A15)+((6/10)*(A15-@INT(A15)))

```

217 fichiers par disquette

Marcel Cottini

Suite au courrier des lecteurs (Pom's 11-12) demandant la manière d'augmenter le nombre de fichiers disponibles sur une disquette, voici un petit programme de 9 secteurs qui répond au problème.

Une disquette DOS 3.3 ne peut stocker plus de 105 programmes ou fichiers dans son catalogue (manuel du DOS, page 123). Pour forcer la main à la routine de formatage, il suffit de lui faire ingurgiter quelques POKES et le tour est joué. Une fois "traitée" par le programme, une disquette voit sa capacité en nombre de fichiers passer de 105 à 217. Comme il ne peut y avoir de création spontanée d'espace disque, le nombre de secteurs disponibles pour les données passe, quant à lui, de 496 à 480.

La VTOC conserve sa place habituelle sur la piste \$11, secteur \$00. Le directory, pour sa part, s'étend non seulement sur les 15 autres secteurs de la piste \$11, mais également sur l'ensemble de la piste \$10 (toujours dans l'ordre décroissant des

secteurs). A raison de 7 noms de fichiers par secteur de catalogue, on peut donc en conserver 31*7 soit 217.

Il n'est pas nécessaire d'initialiser une disquette vierge; une disquette en cours d'utilisation peut être modifiée, pourvu qu'elle n'exploite pas encore la piste \$10 (c'est normalement une des dernières pistes allouées par le DOS). La carte d'occupation (VTOC) est automatiquement mise à jour par le programme ci-joint.

Ce programme nécessite un Apple II (n'importe quelle cuvée) et le DOS 3.3 chargé. Vérifiez bien que ce dernier possède une commande INIT valide (pas de DOS modifié par un des nombreux patches récupérant la place mémoire d'INIT). En cas de doute... Apple vous a fourni une disquette MASTER, utilisez-la !

Fonctionnement du programme

Ligne 80 : routine RWTS "pokée" à l'adresse 768.

Ligne 90 : adresse du tampon utilisé par RWTS.

Ligne 100 : met le volume à zéro.

Ligne 190 à 210 : fixe la variable PK\$ selon le type du programme de boot (Applesoft, binaire ou Exec).

Ligne 230 à 240 : initialisation de la disquette.

Ligne 270 : contrôle si la piste \$10 (16) est déjà occupée, en consultant la VTOC actuelle (piste \$11, secteur \$0).

Ligne 290 : fixe le pointeur vers la piste \$10.

Ligne 300 : pointeur des prochains secteurs.

Ligne 320 à 350 : lecture/écriture des pistes/secteurs par RWTS.

La disquette peut être globalement copiée par n'importe quel programme comme COPYA. Les fichiers peuvent également être copiés individuellement, mais pas par tous les programmes : FID fonctionne correctement, en revanche Super Disk Copy dévoile des lacunes...

```
10 REM -----
20 REM 2 PISTES CATALOG
30 REM PAR MARCEL COTTINI
40 REM COPYRIGHT 84
50 REM -----
70 GOSUB 120
80 POKE 768,32: POKE 769,227: POKE 770,3: POKE
  771,76: POKE 772,217: POKE 773,3: REM SA
  UT A LA ROUTINE RWTS
90 LOC = 10000: POKE 47088,LOC - INT (LOC / 256
  ) * 256: POKE 47089, INT (LOC / 256): REM
  ADRESSAGE MEMOIRE
100 POKE 47083,0: POKE 47091,0: REM MET LE VOLU
  ME A 0
110 RETURN
120 TEXT : HOME : INVERSE : PRINT " 2 PISTES CA
  TALOG ": NORMAL : PRINT : PRINT "UTILISE
  AUSSI LA PISTE 16 COMME CATALOG": PRINT "
  (IL FAUT QU'ELLE SOIT LIBRE)": PRINT
130 FOR X = 1 TO 40: PRINT " ": NEXT X: PRINT
140 VTAB 8: INPUT "VOULEZ-VOUS INITIALISER LE D
  ISK ? REPONDEZ PAR 'OUI' OU 'N' ":
  INS: IF INS = "N" THEN GOTO 250
150 IF INS < > "N" AND INS < > "OUI" THEN 120
160 VTAB 8: CALL - 958: INPUT "NOM DU PROGRAM
  ME DE BOOT: "; NOS: IF NOS < "A" AND NOS >
  "Z" THEN 160
170 VTAB 8: CALL - 958: INPUT "NUMERO VOLUME D
  U DISK: "; VOS: VOL = VAL (VOS): IF VOL <
  1 OR VOL > 254 THEN 170
180 VTAB 8: CALL - 958: INPUT "<A>PPLESOFT
  <B>INAIRE OU
  <T>EXTE ----> ": P
  KS: IF PK$ < > "A" AND PK$ < > "B" AND
  PK$ < > "T" THEN 180
190 IF PK$ = "A" THEN PK$ = "APPLESOFT": POKE 4
  0514,6: GOTO 220
200 IF PK$ = "B" THEN PK$ = "BINAIRE": POKE 405
  14,52: GOTO 220
210 IF PK$ = "T" THEN PK$ = "TEXTE": POKE 40514
```

```
,20: GOTO 220
220 VTAB 8: CALL - 958: HTAB 2: FLASH : PRINT
  "INITIALISATION": NORMAL : PRINT : PRINT
  "PROGRAMME DE BOOT: "; NOS: PRINT "FICHIER
  TYPE.....": PK$: PRINT "NUMERO DE VOLUM
  E. "; VOL
230 PRINT CHR$(4)"INIT" NOS, V VOL
240 PRINT CHR$(4)"DELETE" NOS
250 GOSUB 80
260 VTAB 8: CALL - 958: PRINT "<RETURN> POUR D
  IRECTORY 217 FICHIERS <Q> QUITTER: ":
  GET AS: PRINT AS: IF AS < > CHR$(13)
  THEN END
270 TR = 17: SE = 0: RW = 1: GOSUB 320: IF PEEK (
  LO + 120) = 255 AND PEEK (LO + 121) = 25
  5 THEN POKE LO + 120,0: POKE LO + 121,0:
  RW = 2: GOSUB 320: GOTO 290: REM FIXE LA
  VTOC
280 INVERSE : PRINT : PRINT CHR$(7): " LA PIST
  E 16 EST DEJA UTILISEE !": NORMAL : PRIN
  T : PRINT "(RECOMMENCER AVEC UN AUTRE DIS
  K)": END
290 TR = 17: SE = 1: RW = 1: GOSUB 320: POKE LO +
  1,16: POKE LO + 2,15: RW = 2: GOSUB 320: R
  EM POINTEUR DE LA PISTE 16
300 TR = 16: FOR SE = 15 TO 1 STEP - 1: RW = 1:
  GOSUB 320: POKE LO + 1,16: POKE LO + 2,SE
  - 1: RW = 2: GOSUB 320: NEXT : REM POINTE
  UR DU PROCHAIN SECTEUR
310 PRINT : PRINT "LE DIRECTORY PEUT CONTENIR 2
  17 FICHIERS.": END
320 POKE 47084,TR: POKE 47085,SE: POKE 47092,RW
  : GOSUB 330: CALL 768: RETURN : REM LECTU
  RE ECRITURE SECTEUR
330 IF RW = 1 THEN VTAB 8: CALL - 958: PRINT
  "LECTURE: ";
340 IF RW = 2 THEN VTAB 8: CALL - 958: PRINT
  "ECRITURE: ";
350 PRINT "PISTE-"; TR; ", SECTEUR-"; SE: FOR X =
  1 TO 100: NEXT : RETURN
```


Conversion minuscules/MAJUSCULES

Alexandre Avrane

B/*52!

Oh, excusez-moi, je ne savais que vous n'aviez pas d'affichage en minuscules!

Programmeurs sur Apple //e et //c, ayez un peu de pitié pour les possesseurs d'un "vieux" Apple (1 an et demi!) sans la ROM LC qui permet les minuscules à l'écran. Non qu'ils soient excusables de ne pas se procurer cet accessoire bon marché et si pratique, mais n'oubliez pas qu'un bon programme doit avoir la correction de regarder où il est et ne pas dire "Bonjour" comme ci-dessus...

Deux PEEKs permettent de savoir sur quel type de machine on se trouve:

- PEEK(64255) vaut 0 avec la ROM Autostart, 1 avec l'ancien moni-

teur (Apple II simple ne supportant pas la ROM LC).

- PEEK(64266) vaut 208 pour un Apple II ou II+ avec la ROM Autostart, 240 pour un //e ou //c.

Dans le premier cas (208), il n'existe aucun moyen de savoir s'il est possible d'afficher des minuscules; il faut poser la question, comme le fait Apple Writer II. Si la réponse est négative (pas de minuscules), cela n'oblige pas le programme à modifier toutes ses sorties vidéo, mais plutôt à utiliser une routine de conversion qui intercepte les caractères et les transforme en majuscules.

C'est le but du programme SHIFT, qui convertit également les minuscules accentuées et lettres propres au français; ainsi, Ctrl-X, qui fournit un "backslash" interprété comme un c

cédille par la ROM LC, affiche un C assez surprenant.

Le source (Big Mac) doit être assemblé de préférence à l'adresse \$300. La version pour le programme de BLOC-NOTES, publiée dans le Pom's 13, est logée en \$9A00, c'est-à-dire dans un buffer du DOS inutilisé par le programme, donc avec une sécurité réduite.

Dans la même optique, n'oubliez pas que les Apples ne possèdent pas tous de carte 80 colonnes; il est moins désagréable de lire un message bien poli indiquant qu'il est impossible d'exécuter le programme sur la machine en cours, que de voir son Apple devenir sourd et muet...

```

1 *****
2 *
3 *          SHIFT
4 *
5 *****
6
7 * Copyright [C] 1984  A. Avrane
8
9 * Maj:   02/01/84
10 * Crcc:  02/01/84      Big Mac
11
12 *   Ce programme permet, aux
13 *   utilisateurs d'Apple II+ sans
14 *   rom l.c., de visualiser toutes
15 *   minuscules en majuscules.
16 *   Il se lance par un simple
17 *   BRUN. Reset reconnecte la
18 *   sortie vidéo normale.
19 *   La version assemblée en
20 *   $9A00 se situe au sein des
21 *   buffers du Dos. Prudence donc.
22
23 CSW      =   $36
24 TRAP     =   $48
25 DOSHOOK  =   $3EA
26 COUT0    =   $FDF0
27
28          ORG  $300
29
30          LDA  #<HOOK
31          STA  CSW
32          LDA  #>HOOK
33          STA  CSW+1
34          LDA  #0
35          STA  TRAP
36          JMP  DOSHOOK
37
38 HOOK     STA  CHAR
39          PHP

```

```

40          TXA  sauve les registres
41          PHA
42
43          LDA  CHAR
44          LDX  #7
45 LOOP     CMP  TABLE1,X
46          BEQ  CHANGE
47          DEX
48          BPL  LOOP
49          CMP  #$E0
50          BCC  EXIT
51          SEC
52          SBC  #$20
53          BNE  EXIT      =jmp
54 CHANGE   LDA  TABLE2,X
55 EXIT     STA  CHAR
56          PLA
57          TAX
58          PLP
59          LDA  CHAR
60          JMP  COUT0      video
61 TABLE1  DFB  $C0      ;@
62          DFB  $DC      ;\
63          DFB  $E0
64          DFB  $FB      ;(
65          DFB  $FC      ;!
66          DFB  $FD      ;)
67          DFB  $FE      ;~
68          DFB  $FF      ;
69 TABLE2  ASC  "A"
70          ASC  "C"
71          ASC  "!" + $20
72          ASC  "E" + $20
73          ASC  "U" + $20
74          ASC  "E" + $20
75          ASC  "-" + $20
76          ASC  "!" + $20
77 CHAR     DFB  0
78          END

```

Récapitulation SHIFT

0300- A9 0F 85 36 A9 03 85 37
0308- A9 00 85 48 4C EA 03 8D

0310- 4A 03 08 8A 48 AD 4A 03
0318- A2 07 DD 3A 03 F0 0C CA
0320- 10 F8 C9 E0 90 08 38 E9
0328- 20 D0 03 BD 42 03 8D 4A

0330- 03 68 AA 28 AD 4A 03 4C
0338- F0 FD C0 DC E0 FB FC FD
0340- FE FF C1 C3 A1 C5 D5 C5
0348- AD A1 00

Un catalogue général

Michel Baudrand

Qui n'a jamais souhaité retrouver la disquette sur laquelle est stocké tel fichier, ou l'adresse de chargement d'un programme assembleur ? CATGEN répond à ce souci et vous permet en effet de :

- stocker le catalogue d'un ensemble de disquettes dans un seul fichier;
- trier ces informations selon différents critères;
- éditer le résultat d'un tri à l'écran ou sur imprimante.

CATGEN fonctionne sur Apple II+ ou IIe, avec 1 ou 2 lecteurs de disquettes et supporte une imprimante. Il est écrit en Basic et utilise deux routines en assembleur : l'accès à la routine RWTS pour la lecture du catalogue et un programme de tri.

Après avoir demandé la date et s'être assuré de sa validité, CATGEN propose un menu de 8 rubriques :

1 - INITIALISATIONS : permet de définir le nombre de lecteurs (1 par défaut) et le type de l'imprimante (en fait EPSON ou autres). C'est là qu'est initialisé le fichier de données baptisé "CATALOGUE GENERAL".

2 - SAISIE de disquettes DOS 3.3 : le programme demande s'il s'agit d'une mise à jour par la question "EST-CE UNE NOUVELLE DISQUETTE (O/N) ? ", puis les caractéristiques de la disquette traitée : NUMERO, FACE, LABEL, CLASSE des fichiers.

Le programme lit et analyse la piste 17 (\$11) de la disquette pour déterminer les caractéristiques de chaque fichier, à savoir : le NOM, le TYPE, la TAILLE en secteurs, l'ADRESSE de CHARGEMENT et la LONGUEUR des programmes binaires.

Un dialogue permet de compléter les informations : CLASSE (numéro de 1 à 16 dont la signification est définie par l'utilisateur), NOM d'un fichier lié (fichier de la même disquette associé au fichier en cours).

3 - SAISIE de disquettes "SPECIALE" : il s'agit de disquettes Type Pascal ou non "cataloguables". Les informations correspondant à chaque fichier sont saisies par dialogue au clavier.

4 - COMPACTAGE du fichier CATALOGUE GENERAL : il doit être

effectué après une mise à jour.

5 - TRI : un menu offre le choix entre 5 critères de tri : par Nom, par Numéro de Disquette, par Label, par Type, par Classe. Le résultat du tri peut être sauvegardé dans le fichier TRI. Pour chacun de ces critères, les fichiers sont classés par ordre alphabétique.

6 - EDITION : après définition de l'organe de sortie (Ecran ou Imprimante), un menu propose 5 formats d'édition. Le choix 5 est adapté à l'Epson MX82. Une édition n'est faite qu'après un tri et suppose l'existence du fichier TRI.

7 - MODE D'EMPLOI : une description succincte des possibilités de CATGEN est affichée.

8 - Enfin, le choix 8 permet de quitter CATGEN.

A la fin de chaque session, il est possible, soit de revenir au MENU, soit de reprendre la même session.

CATGEN est constitué de 5 programmes implantés sur une disquette placée dans l'unité 1 et appelée "disquette programme". Le fichier de données CATALOGUE GENERAL est sur la même disquette.

Pour une configuration à 2 lecteurs, la disquette analysée est dans l'unité 2.

Des messages permettent de s'assurer à chaque opération que la disquette installée est la bonne.

Sauf spécification explicite, les réponses à donner sous forme d'un seul caractère ne nécessitent pas le 'RETURN'.

Quelques remarques sont nécessaires en ce qui concerne la phase de saisie :

- Pour alléger le dialogue de validation des informations, il faut répondre "O" à la question "EST-ELLE GLOBALE (O/N) ?" pour la classe des fichiers.
- Par contre, pour filtrer certains fichiers (ne pas les inclure dans le CATALOGUE GENERAL), il convient de répondre "N" à cette même question et d'utiliser ensuite la touche ESC lorsque la question "OK (O/N OU 'ESC') ?" est

posée.

- Pour mettre à jour les informations concernant une disquette, il faut répondre "N" à la question "EST-CE UNE NOUVELLE DISQUETTE (O/N) ?".
- Le nom du fichier lié doit appartenir à la même disquette ou être absent.
- Si on répond 'RETURN' au paramètre CLASSE, la valeur prise est celle définie le cas échéant dans la phase de définition de la disquette, sinon il s'agit de celle du fichier précédemment traité.

On peut enchaîner la SAISIE de plusieurs disquettes avant d'effectuer la mise à jour du fichier CATALOGUE GENERAL, et ce jusqu'à concurrence de 100 fichiers. A la question "ENCORE (O/N) ?" il suffit de répondre "O". Le nombre de fichiers saisis est affiché en "%" (TAUX).

La signification du paramètre CLASSE peut être personnalisée. Il suffit de redéfinir 16 rubriques dans les instructions DATA en fin du programme CATGEN.SAISIE (après DATA RUB - lignes 2980 à 3000).

L'opération de COMPACTAGE se fait par recopie du fichier CATALOGUE GENERAL sur la disquette programme. Ceci limite la taille du fichier de données à 500 articles environ. Par contre, le tri s'accommodant d'un fichier non compacté.

En cas de détection d'erreur, le programme concerné s'arrête après édition du numéro de la ligne erronée, sauf dans CATGEN où le MENU est relancé.

Quelques précisions techniques

Le CATALOG est lu par appel à RWTS et analysé en Basic. L'appel à RWTS est implanté de 768 à 798. Le buffer lu est en \$9400.

La routine de tri (fichier SPEED SORT), empruntée à NIBBLE, est accolée en fin de CATGEN.TRI par modification du pointeur \$AF-\$B0. Le tri est effectué en mémoire, d'où la limitation à 1000 du nombre de fichiers gérés.

Les mémoires 800 à 811 servent à la communication entre les divers programmes (date, nombre de drives, type d'imprimante).

```

100 REM *****
110 REM * CATALOGUE.. *
120 REM * GENERAL: MENU *
130 REM * JUIN 1984 *
140 REM *****
150 ONERR GOTO 690
160 DS = CHR$(4)
170 FS = "CATALOGUE GENERAL"
180 RDS = DS + "READ "
190 WRS = DS + "WRITE "
200 OPS = DS + "OPEN "
210 CLS = DS + "CLOSE "
220 DLS = DS + "DELETE"
230 REM *****
240 REM * MENU *
250 REM *****
260 TEXT : HOME
270 HOME : INVERSE : PRINT "DATE";: NOR
MAL : PRINT " ";: DAS;
280 HTAB 30: PRINT "CATGEN ";: INVERSE
: PRINT "V2.1";: NORMAL
290 FOR I = 1 TO 40: PRINT " ";: NEXT I
300 REM LECTURE DE LA DATE
310 IF PEEK(800) < > 255 THEN 350
320 DAS = ""
330 FOR I = 1 TO 8: DAS = DAS + CHR$(
PEEK(800 + I)): NEXT I
340 GOTO 400
350 VTAB 10: INPUT "DATE (JJ/MM/AA) : "
: DAS
352 JJS = "": MMS = ""
353 FOR I = 1 TO LEN(DAS)
354 JS = MID$(DAS,I,1): IF JS = "/" OR
JS = " " THEN I1 = I: GOTO 357
355 JJS = JJS + JS
356 NEXT I
357 FOR I = I1 + 1 TO LEN(DAS) - I1
358 MS = MID$(DAS,I,1): IF MS = "/" OR
MS = " " THEN 361
359 MMS = MMS + MS
360 NEXT I
361 ANS = RIGHT$(DAS,2)
362 AA = VAL(ANS): MM = VAL(MMS): JJ =
VAL(JJS)
363 IF JJ < 1 OR JJ > 31 OR MM < 1 OR M
M > 12 OR AA < 84 THEN PRINT CH
R$(7): GOTO 350
364 IF LEN(JJS) = 1 THEN JJS = "0" +
JJS
365 IF LEN(MMS) = 1 THEN MMS = "0" +
MMS
366 DAS = JJS + "/" + MMS + "/" + ANS
370 FOR I = 1 TO 8: POKE 800 + I, ASC(
MID$(DAS,I,1)): NEXT I
380 POKE 34,4: HOME : POKE 34,1
385 POKE 810,1: POKE 811,1: REM PAR DE
FAUT '1 DRIVE'
390 POKE 800,255: REM DRAPEAU POUR L'I
NITIALISATION
400 VTAB 1: HTAB 6: PRINT DAS
410 POKE 34,2
420 HOME : HTAB 17: INVERSE : PRINT "M
E N U": NORMAL : PRINT
430 HTAB 4: INVERSE : PRINT "1";: NORMA
L : PRINT "-INITIALISATIONS"
440 PRINT
450 HTAB 4: INVERSE : PRINT "2";: NORMA

```

```

L : PRINT "-DISQUETTE STANDARD DO
S 3.3"
460 PRINT
470 HTAB 4: INVERSE : PRINT "3";: NORMA
L : PRINT "-DISQUETTE 'SPECIALE'"
480 PRINT
490 HTAB 4: INVERSE : PRINT "4";: NORMA
L : PRINT "-COMPACTAGE DU FICHIER
FS"
500 PRINT
510 HTAB 4: INVERSE : PRINT "5";: NORMA
L : PRINT "-TRI..."
520 PRINT
530 HTAB 4: INVERSE : PRINT "6";: NORMA
L : PRINT "-EDITION.."
540 PRINT
550 HTAB 4: INVERSE : PRINT "7";: NORMA
L : PRINT "-MODE D'EMPLOI.."
560 PRINT
570 HTAB 4: INVERSE : PRINT "8";: NORMA
L : PRINT "-FIN DE SESSION"
580 VTAB 22: INVERSE : PRINT "CHOISISSE
Z VOTRE SESSION (1..8)": NORMAL
: PRINT " ";: GET Z$: PRINT Z$: ME
= VAL(Z$)
590 IF ME < 1 OR ME > 8 THEN 580
600 POKE 809,ME
610 ON ME GOSUB 700,910,910,1040,940,98
0,1010,660
620 Z = FRE(0)
630 VTAB 23: CALL - 958: INVERSE : PRI
NT "RETOUR AU MENU (O/N)?": NORM
AL : PRINT " ";: GET Z$: PRINT Z$
640 IF Z$ = "N" THEN 660
650 GOTO 420
660 TEXT : HOME : PRINT CLS: VTAB 10: H
TAB 14: PRINT "AU REVOIR...": VTA
B 23
670 END
680 REM SI 'ERREUR',RELANCE DU MENU
690 PRINT DS: PRINT DS"PREO": GOTO 240
700 REM * INITIALISATIONS" *
710 HOME
720 HTAB 12: INVERSE : PRINT "INITIALIS
ATIONS": NORMAL
730 VTAB 5: PRINT "CONFIGURATION AVEC "
: INVERSE : PRINT "1";: NORMAL :
PRINT " OU ";: INVERSE : PRINT "
2";: NORMAL : PRINT " DISQUES ?
": GET DKS: PRINT DKS
740 DK = VAL(DK$): IF DK < > 1 AND DK
< > 2 THEN 730
750 POKE 810,DK
760 VTAB 7: PRINT "L'IMPRIMANTE EST-ELL
E UNE EPSON("): INVERSE : PRINT "
O/N";: NORMAL : PRINT ")?": GET
TYPLP$: PRINT TYPLP$
770 IF TYPLP$ < > "O" AND TYPLP$ < >
"N" THEN 760
780 POKE 811,80: IF TYPLP$ = "N" THEN
POKE 811,1
790 VTAB 9: PRINT "CREATION DU FICHIER
("): INVERSE : PRINT "O/N";: NORM
AL : PRINT " + ";: INVERSE : PRIN
T "RETURN";: NORMAL : PRINT " ) "
: INPUT Z$
800 IF Z$ = "N" THEN RETURN
810 IF Z$ < > "O" THEN 790
820 REM NOX=NOMBRE MAXI D'ARTICLES
830 REM LOX=TAILLE DES ARTICLES

```



```

840 NO% = 1:LO% = 87
850 VTAB 14: FLASH : PRINT "CREATION";:
      NORMAL : PRINT " DU FICHIER ";:
      INVERSE : PRINT F$: NORMAL
860 PRINT OP$F$: PRINT DL$F$: PRINT OP$
      F$
870 PRINT WR$F$;" ,RO"
880 PRINT NO%: PRINT LO%
890 PRINT CL$
900 RETURN
910 REM * DISQUETTE STANDARD 3.3 OU 'SP
      ECIALE' *
920 CC$ = "CATGEN.SAISIE"
930 GOTO 960
940 REM * TRI *
950 CC$ = "CATGEN.TRI"
960 PRINT D$"RUN "CC$
970 RETURN
980 REM * EDITION *
990 CC$ = "CATGEN.EDIT"
1000 GOTO 960
1010 REM * MODE D'EMPLOI *
1020 CC$ = "CATGEN.UTILISATION"
1030 GOTO 960
1040 REM * COMPACTAGE DE F$ *
1050 HOME : HTAB 14: INVERSE : PRINT "C
      OMPACTAGE": NORMAL
1060 VTAB 12
1070 FLASH : PRINT "COMPACTAGE";: NORMA
      L : PRINT " DU FICHIER ";: INVERS
      E : PRINT F$: NORMAL
1080 F1$ = "TRAV001"
1090 PRINT OP$F$
1100 PRINT RD$F$;" ,RO"
1110 INPUT NO%: INPUT LO%
1120 PRINT CL$: IF NO% = 1 THEN 1320
1130 PRINT OP$F$;" ,L":LO%
1140 PRINT OP$F1$;" ,L":LO%
1150 N1% = 0
1160 FOR I = 1 TO NO% - 1
1170 PRINT RD$F$;" ,R" I
1180 INPUT ST$
1190 IF MID$(ST$,5,1) = " " THEN 1250
1200 N1% = N1% + 1
1210 PRINT WR$F1$;" ,R"N1%
1220 N1$ = STR$(N1%): FOR N = LEN (N1
      $) TO 3:N1$ = "0" + N1$: NEXT N
1230 ST$ = CHR$(34) + N1$ + " " + RIG
      HT$(ST$,LO% - 7)
1240 PRINT ST$
1250 NEXT I
1260 PRINT WR$F1$;" ,RO"
1270 PRINT N1% + 1: PRINT LO%
1280 PRINT CL$F1$
1290 PRINT CL$F$
1300 PRINT D$"DELETE"F$
1310 PRINT D$"RENAME"F1$,"F$
1320 PRINT : PRINT "IL Y A ";: INVERSE
      : PRINT N1%: NORMAL : PRINT "
      ARTICLES ."
1330 RETURN

```

```

100 REM *****
110 REM * CATALOGUE.. *
120 REM * GENERAL: EDIT *
130 REM *****
135 HIMEM: 38400

```

```

140 ONERR GOTO 1330
150 DIM ST$(1000)
170 ES$ = CHR$(27)
180 HT$ = " "
190 TB$ = CHR$(9)
200 D$ = CHR$(4)
210 F$ = "CATALOGUE GENERAL"
220 RD$ = D$ + "READ "
230 WR$ = D$ + "WRITE "
240 OP$ = D$ + "OPEN "
250 CL$ = D$ + "CLOSE "
260 DL$ = D$ + "DELETE"
270 TR$ = "TRI"
280 REM LECTURE DE LA DATE
290 FOR I = 1 TO 8
300 DA$ = DA$ + CHR$( PEEK (800 + I))
310 NEXT I
330 GOTO 470
340 REM DECOMPACTAGE DE ST$
350 ST$ = RIGHT$(ST$, LEN (ST$) - 5)
360 P1$ = MID$(ST$,1,5): REM ND$+F$
370 P2$ = MID$(ST$,6,20): REM NA$
380 P3$ = MID$(ST$,39,20): REM FICH.
      LIEN
390 P4$ = MID$(ST$,59,2): REM CLASSE
400 P5$ = MID$(ST$,61,8): REM DATE
410 P6$ = MID$(ST$,26,2): REM TYPE
420 P7$ = MID$(ST$,28,3): REM TAILLE
430 P8$ = MID$(ST$,31,4): REM ADRESSE
440 P9$ = MID$(ST$,35,4): REM LONGUEU
      R
450 PA$ = MID$(ST$,69,12): REM LABEL
460 RETURN
470 REM EDITION
480 HOME : HTAB 17: INVERSE : PRINT "ED
      ITION": NORMAL
490 POKE 34,4
500 VTAB 5: PRINT "SUR ";: INVERSE : PR
      INT "(E)": NORMAL : PRINT "CRAN
      OU ";: INVERSE : PRINT "(I)": NO
      RMAL : PRINT "MPRIMANTE ? ";: GET
      Z$: PRINT Z$
510 IF Z$ < > "E" AND Z$ < > "I" THEN
      500
520 IF Z$ = "E" THEN TERM = 0
530 IF Z$ = "I" THEN TERM = 1
540 REM LIT LE FICHIER 'TRI'
550 PRINT OP$TR$: PRINT RD$TR$
560 INPUT N1%
570 FOR I = 1 TO N1%
580 INPUT ST$(I)
590 NEXT I
600 PRINT CL$
610 PRINT
620 INVERSE : PRINT "1";: NORMAL : PRIN
      T "-STANDARD :FIC,TYP,TAIL,AD
      R,LNG,NDK,LAB"
630 PRINT
640 INVERSE : PRINT "2";: NORMAL : PRIN
      T "-PAR DISQUETTE:NDK,FIC,TYP,TAI
      L,ADR,LNG,LAB"
650 PRINT
660 INVERSE : PRINT "3";: NORMAL : PRIN
      T "-PAR CLASSE :CLA,FIC,TYP,TAI
      L,ADR,LNG,NDK,LAB"
670 PRINT
680 INVERSE : PRINT "4";: NORMAL : PRIN
      T "-PAR TYPE :TYP,FIC,TAIL,AD
      R,LNG,NDK,LAB"
690 PRINT

```

```

700 INVERSE : PRINT "5";: NORMAL : PRIN
    T "-COMPLET :NDK,FIC,TYP,TAI
    L,ADR,LNG,LIE,CLA,DATE"
710 VTAB 22: INVERSE : PRINT "CHOISISSE
    Z VOTRE FORMAT (1..5) ?";: NORMAL
    : PRINT " ";: GET Z$: PRINT Z$
720 ZZ = VAL (Z$): IF ZZ < 1 OR ZZ > 5
    THEN 710
730 IF ZZ = 5 AND PEEK (811) < > 80 T
    HEN 710
740 REM OUVRE F$
750 PRINT OP$F$
760 PRINT RD$F$;" ,RO": INPUT NO$: INPUT
    LO$
770 PRINT CL$F$
780 PRINT OP$F$;" ,L";LO$
784 VTAB 5: CALL - 958: VTAB 22: PRINT
    "<TAPER 'ESP' POUR SUSPENDRE L'E
    DITION>": PRINT "<PUIS UNE 2EME F
    OIS POUR LA REPRENDRE>";
786 POKE 35,21: VTAB 4: HOME : PRINT
790 IF TERM = 0 THEN 850
800 INPUT "CADRER LE PAPIER ET TAPER 'R
    ETURN' ? ";Z$
810 PRINT D$"PRE1": IF ZZ < > 5 THEN
    PRINT ES$;"Q"; CHR$ (80); CHR$ (1
    8);
820 IF ZZ = 5 THEN POKE 1657,110: PRIN
    T ES$;"Q"; CHR$ (110); CHR$ (15);
    : PRINT ES$;"D"; CHR$ (7); CHR$ (
    28); CHR$ (34); CHR$ (41); CHR$ (
    49); CHR$ (55); CHR$ (62); CHR$ (
    84); CHR$ (89); CHR$ (0);
830 PRINT :NN = 0
840 GOSUB 1070: REM TITRE
850 FOR I = 1 TO N1$
870 N2$ = ST$(I) - 1: IF N2$ = 0 THEN 96
    0
880 LAST$ = ST$
890 PRINT RD$F$;" ,R";N2$
900 INPUT ST$
910 GOSUB 340: REM DECOMPACTAGE
920 GOSUB 1200: REM EDITION
921 PRINT D$
922 KEY = PEEK ( - 16384): POKE - 1636
    8,0: IF KEY < 127 THEN 930
924 IF KEY = 128 + 32 THEN GET Z$: PRI
    NT Z$
930 IF TERM = 0 THEN 960
940 NN = NN + 1
950 IF NN = 58 THEN FOR J = 1 TO 6: PR
    INT : NEXT J:NN = 0: GOSUB 1070
960 NEXT I
970 IF TERM = 0 THEN PRINT D$: GOTO 99
    0
980 PRINT ES$;"a": PRINT D$;"PRE0"
990 PRINT
1000 POKE 35,23: VTAB 21: CALL - 958:
    VTAB 23
1010 INVERSE : PRINT "RETOUR AU MENU (0
    /N) ?";: NORMAL : PRINT " ";: GET
    Z$: PRINT Z$;
1020 IF Z$ = "N" THEN 470: REM ON REBO
    UCLE
1030 CC$ = "CATGEN"
1050 PRINT D$"RUN"CC$
1060 STOP
1070 REM TITRE DE CHAQUE PAGE
1080 PRINT "DATE: ";DAS$: HTAB 34: PRIN
    T "CATALOGUE GENERAL": PRINT

```

```

1090 ON ZZ GOTO 1100,1120,1140,1160,118
    0
1100 PRINT " ";:"NOM DE FICHIER
    ";
    ;"TYPE";" TAIL. ";"ADR.";HT$;"LNG
    .";HT$;"DISK.";HT$;"LABEL"
1110 FOR I1 = 1 TO 64: PRINT "-";: NEXT
    I1: PRINT : RETURN
1120 PRINT HT$;"DISK.";HT$;"NOM DE FICH
    IER
    ";:"TYPE" TAIL. ";"ADR.";H
    T$;"LNG."; " LABEL"
1130 FOR I1 = 1 TO 63: PRINT "-";: NEXT
    I1: PRINT : RETURN
1140 PRINT ;"CLAS";HT$;"NOM DE FICHIER
    ";:"TYPE";" TAIL."; "ADR.";HT
    $;"LNG.";HT$;"DISK.";HT$;"LABEL"
1150 FOR I1 = 1 TO 68: PRINT "-";: NEXT
    I1: PRINT : RETURN
1160 PRINT "TYPE";HT$;"NOM DE FICHIER
    TAIL. ADR.";HT$;"LNG.";HT$;"
    DISK. "; "CLAS";" LABEL"
1170 FOR I1 = 1 TO 68: PRINT "-";: NEXT
    I1: PRINT : RETURN
1180 PRINT "DISK.";TB$;"NOM DE FICHIER"
    ;TB$;"TYPE";TB$;"TAILLE";TB$;"ADR
    .";TB$;"LNG.";TB$;"FICHIER LIEN"
    ;TB$;"CLAS";TB$;"DATE";" LABE
    L"
1190 FOR I1 = 1 TO 109: PRINT "-";: NEX
    T I1: PRINT : RETURN
1200 REM EDITION
1210 ON ZZ GOTO 1230,1250,1270,1290,131
    0
1220 GOTO 1410
1230 IF LEFT$(P2$,1) < > MID$(LAST
    $,6,1) THEN PRINT :I = I - 1: RE
    TURN
1240 PRINT HT$:P2$;HT$:P6$;HT$:P7$;HT$:
    P8$;HT$:P9$;HT$:P1$;HT$:PA$: RETU
    RN
1250 IF P1$ < > MID$(LAST$,1,5) THEN
    PRINT :I = I - 1: RETURN
1260 PRINT HT$:P1$;HT$:P2$;HT$:P6$;HT$:
    P7$;HT$:P8$;HT$:P9$;HT$:PA$: RETU
    RN
1270 IF P4$ < > MID$(LAST$,59,2) THE
    N PRINT :I = I - 1: RETURN
1280 PRINT HT$:P4$;HT$:P2$;HT$:P6$;HT$:
    P7$;HT$:P8$;HT$:P9$;HT$:P1$;HT$:P
    A$: RETURN
1290 IF RIGHT$(P6$,1) < > MID$(LAS
    T$,27,1) THEN PRINT :I = I - 1:
    RETURN
1300 PRINT HT$:P6$;HT$:P2$;HT$:P7$;HT$:
    P8$;HT$:P9$;HT$:P1$;HT$:P4$;HT$:P
    A$: RETURN
1310 IF P1$ < > MID$(LAST$,1,5) THEN
    PRINT :I = I - 1: RETURN
1320 PRINT P1$;TB$;P2$;TB$;P6$;TB$;P7$;
    TB$;P8$;TB$;P9$;TB$;P3$;TB$;P4$;T
    B$;P5$;HT$:PA$: RETURN
1330 PRINT D$: PRINT "ERREUR "; PEEK (2
    22);" LIGNE "; PEEK (218) + PEEK
    (219) * 256: STOP

```

```

100 REM *****
110 REM * CATALOGUE.. *

```

```

120 REM * GENERAL:UTILISATION*
130 REM *****
140 ONERR GOTO 580
150 D$ = CHR$(4)
160 REM MODE D'EMPLOI
170 HOME : HTAB 12
180 INVERSE : PRINT "CATGEN";: NORMAL :
    PRINT " PERMET DE : "
190 PRINT
200 PRINT " 1-";: INVERSE : PRINT "CREE
    R";: NORMAL : PRINT " LE CATALOGU
    E D'UN ENSEMBLE DE DISQUETTES"
210 PRINT " 2-";: INVERSE : PRINT "TRIE
    R";: NORMAL : PRINT " LES ELEMENT
    S DE CE CATALOGUE"
220 PRINT " 3-";: INVERSE : PRINT "EDIT
    ER";: NORMAL : PRINT " LES RESULT
    ATS DU TRI"
230 PRINT " 4-";: INVERSE : PRINT "REAL
    ISER";: NORMAL : PRINT " LA MISE
    A JOUR DE CES ": PRINT "INFORMATI
    ONS"
240 PRINT
250 INVERSE : PRINT "CATGEN";: NORMAL :
    PRINT " LIT LE CATALOGUE DES DIS
    QUETTES DITES 'STANDARD' (DOS 3,
    3) ET "
260 PRINT "PERMET LA SAISIE DES DONNEES
    CORRESPON-";: PRINT "DANTES POUR
    LES DISQUETTES 'SPECIALES' (NON
    DOS 3.3 OU PROTEGEES)"
270 PRINT
280 INVERSE : PRINT "CATGEN";: NORMAL :
    PRINT " TRAVAILLE SUR UN APPLE I
    I 48K , 10U 2 LECTEURS , SUPPORTE
    UNE IMPRIMANTE": PRINT "ET PERME
    T DE GERER 1000 FICHIERS ENVIRON"
290 GOSUB 590
300 PRINT " LES INFORMATIONS MEMORISEES
    DANS LE"
310 PRINT "FICHIER ";: INVERSE : PRINT
    "CATALOGUE GENERAL";: NORMAL : PR
    INT " SONT : "
320 PRINT
330 PRINT " 1-LE ";: INVERSE : PRINT "N
    Umero";: NORMAL : PRINT " DE LA D
    ISQUETTE: 4 CARAC-": PRINT "TERES
    ET LE ";: INVERSE : PRINT "CODE"
    ;: NORMAL : PRINT " DE LA FACE 'A
    ' OU 'B'"
340 PRINT " 2-LE ";: INVERSE : PRINT "N
    OM";: NORMAL : PRINT " DU FICHIER
    ,LIMITE A 20 CARAC-": PRINT "TERE
    S"
350 PRINT " 3-LE ";: INVERSE : PRINT "T
    YPE";: NORMAL : PRINT " DU FICHIE
    R: 'A','B','I','R','T' ET SON E
    TAT 'LOCK' OU 'UNLOCK'"
360 PRINT " 4-LA ";: INVERSE : PRINT "T
    AILLE";: NORMAL : PRINT " DU FICH
    IER EN SECTEURS"
370 PRINT " 5-L'";: INVERSE : PRINT "AD
    RESSE";: NORMAL : PRINT " DE CHAR
    GEMENT POUR LES PRO-GRAMMES BINA
    IRES"
380 PRINT " 6-LA ";: INVERSE : PRINT "L
    ONGUEUR";: NORMAL : PRINT " DES P
    ROGRAMMES BINAIREs"
390 PRINT " 7-LE ";: INVERSE : PRINT "N
    OM";: NORMAL : PRINT " D'UN FICHI

```

```

ER ":: INVERSE : PRINT "LIE";: NO
    RMAL : PRINT " :FICHIER DE LA ME
    ME DISQUETTE ASSOCIE AU FICHIER E
    N COURS"
400 PRINT " 8-LA ";: INVERSE : PRINT "C
    LASSE";: NORMAL : PRINT " DU PROG
    RAMME,CHOISIE DANS": PRINT "UNE L
    ISTE PERSONNALISABLE"
410 PRINT " 9-LA ";: INVERSE : PRINT "D
    ATE";: NORMAL : PRINT " DU JOUR"
420 PRINT "10-LE ";: INVERSE : PRINT "L
    ABEL";: NORMAL : PRINT " DE LA DI
    SQUETTE"
430 GOSUB 590
440 PRINT " L'ENCHAINEMENT DES OPERATIO
    NS EST : "
450 PRINT
460 PRINT " -";: INVERSE : PRINT "INIT
    IALISATIONS";: NORMAL : PRINT " :
    DU FICHIER 'CATALOGUE GENERAL' ,
    DU NOMBRE DE LECTEURS , DU TYPE D
    'IMPRIMANTE (CHOIX 1)"
470 PRINT " -";: INVERSE : PRINT "SAIS
    IE";: NORMAL : PRINT " OU ";: INV
    ERSE : PRINT "MISE A JOUR";: NORM
    AL : PRINT " DES FICHIERS": PRINT
    "(CHOIX 2 OU 3 SELON LA DISQUETT
    E)"
480 PRINT " -";: INVERSE : PRINT "DEFI
    NITION DES PARAMETRES DU TRI";: N
    ORMAL : PRINT " ET TRI (CHOIX 5
    );LE RESULTAT EST STOCKE ": PRINT
    "DANS LE FICHIER ";: INVERSE : P
    RINT "TRI": NORMAL
490 PRINT " -";: INVERSE : PRINT "DEF
    INITION DU FORMAT D'EDITION";: NO
    RMAL : PRINT " PUIS EDITION SUR L
    E 'TERMINAL' OU SUR L'IMPRIMANTE'
    (CHOIX 6)"
500 PRINT " -LE ";: INVERSE : PRINT "C
    OMPACTAGE";: NORMAL : PRINT " DU
    FICHIER (CHOIX 4)": PRINT "EST A
    FAIRE APRES UNE OPERATION DE MISE
    A JOUR"
502 INVERSE : PRINT "N.B.";: NORMAL : P
    RINT "S'IL Y A 2 DISQUES,LES DISQ
    UETTES A": PRINT "ANALYSER SERONT
    MISES DANS L'UNITE ";: INVERSE :
    PRINT 2: NORMAL
504 PRINT "L'UNITE ";: INVERSE : PRINT
    1;: NORMAL : PRINT " SUPPORTANT "
    ;: INVERSE : PRINT "CATGEN";: NOR
    MAL : PRINT " ET LES"
506 PRINT "DONNEES";: INVERSE : PRINT
    "CATALOGUE GENERAL";: NORMAL : PR
    INT " ET ";: INVERSE : PRINT "TRI
    ": NORMAL
510 VTAB 23
520 INVERSE : PRINT "RETOUR AU MENU (O/
    N) ?";: NORMAL : PRINT " ";: GET
    Z$: PRINT Z$
530 IF Z$ = "N" THEN END
540 CC$ = "CATGEN"
560 PRINT D$:"RUN"CC$
570 STOP
580 PRINT D$: PRINT "ERREUR "; PEEK (22
    2);" LIGNE "; PEEK (218) + PEEK
    (219) * 256: STOP
590 VTAB 23: INVERSE : PRINT "TAPER UNE
    TOUCHE POUR CONTINUER";: NORMAL

```



```

      : PRINT " ";: GET Z$: PRINT Z$
600 HOME
610 RETURN

100 REM *****
110 REM * CATALOGUE.. *
120 REM * GENERAL:TRI *
130 REM *****
140 HIMEM: 38400
150 ONERR GOTO 910
160 DIM ST$(1000)
170 DIM S$(1000)
180 BS = CHR$(7)
190 DS = CHR$(4)
200 FS = "CATALOGUE GENERAL"
210 RDS = DS + "READ "
220 WRS = DS + "WRITE "
230 OPS = DS + "OPEN "
240 CLS = DS + "CLOSE "
250 DLS = DS + "DELETE"
260 TR$ = "TRI"
270 REM TRI...
280 HOME : VTAB 3
290 HTAB 12: INVERSE : PRINT "DEFINITIO
    N DU TRI": NORMAL : PRINT
300 POKE 34,4
310 PRINT : INVERSE : PRINT "1": NORMA
    L : PRINT "-PAR NOM DE FICHIER"
320 PRINT : INVERSE : PRINT "2": NORMA
    L : PRINT "-PAR NUMERO DE DISQUET
    TE"
330 PRINT : INVERSE : PRINT "3": NORMA
    L : PRINT "-PAR LABEL DE DISQUETT
    E"
340 PRINT : INVERSE : PRINT "4": NORMA
    L : PRINT "-PAR CLASSE DE PROGRAM
    ME"
350 PRINT : INVERSE : PRINT "5": NORMA
    L : PRINT "-PAR TYPE DE FICHIER"
360 VTAB 22: INVERSE : PRINT "LEQUEL (1
    ..5) ?": NORMAL : PRINT " ";: GE
    T Z$: PRINT Z$: TT = VAL (Z$)
370 IF TT < 1 OR TT > 5 THEN 360
380 REM LECTURE DE FS DANS ST$
390 VTAB 18: FLASH : PRINT "LECTURE":
    NORMAL : PRINT " DU FICHIER ": I
    NVERSE : PRINT FS: NORMAL
400 REM OUVRE FS
410 PRINT OPSFS
420 PRINT RDSFS;" ,RO": INPUT NO$: INPUT
    LO$
430 PRINT CLSFS
440 PRINT OPSFS;" ,L": LO$
450 ST$(0) = " "
460 N1% = 0
470 FOR I = 1 TO NO% - 1
480 PRINT RDSFS;" ,R": I
490 INPUT ST$

```

```

500 IF MID$(ST$,5,1) = "*" THEN 630
510 N1% = N1% + 1
520 ON TT GOTO 530,550,570,590,610
530 ST$(N1%) = MID$(ST$,11,20) + MID$
    (ST$,6,5)
540 GOTO 620
550 ST$(N1%) = MID$(ST$,6,25)
560 GOTO 620
570 ST$(N1%) = MID$(ST$,69,12) + MID$
    (ST$,11,20)
580 GOTO 620
590 ST$(N1%) = MID$(ST$,64,2) + MID$
    (ST$,11,20)
600 GOTO 620
610 ST$(N1%) = MID$(ST$,32,1) + MID$
    (ST$,11,20)
620 HTAB 9: VTAB 20: PRINT D$: PRINT "A
    RTICLE ";I
630 NEXT I
640 PRINT CLSFS
650 HOME : FLASH : VTAB 10: HTAB 16: PR
    INT " T R I ": NORMAL
660 X = FRE (0)
670 PRINT BS
675 REM APPEL DE LA ROUTINE DE TRI
680 POKE 964,(N1% / 256 - INT (N1% / 2
    56)) * 256
690 POKE 965, INT (N1% / 256)
700 CALL PEEK (175) + PEEK (176) * 25
    6 - 261ST$(0),S$(0)
710 PRINT BS
720 VTAB 12
730 PRINT "SAUVEGARDE DU RESULTAT DU TR
    I (O/N) ? ": GET Z$: PRINT Z$
740 IF Z$ = "N" THEN 870
750 IF Z$ < > "O" THEN 720
760 PRINT
770 FLASH : PRINT "ECriture": NORMAL :
    PRINT " DANS LE FICHIER ": INVE
    RSE : PRINT TR$: NORMAL
780 TR$ = "TRI": PRINT OPSTRS: PRINT DLS
    TR$: PRINT OPSTR$: PRINT WRSTR$
790 PRINT N1%
800 FOR I = 1 TO N1%
820 IF S$(I) = 0 THEN N% = LAST%: GOTO
    840
830 N% = S$(I):LAST% = N%
840 PRINT N% + 1
850 NEXT I
860 PRINT CLS
870 VTAB 23: INVERSE : PRINT "RETOUR AU
    MENU (O/N) ?": NORMAL : PRINT "
    ": GET Z$: PRINT Z$
880 IF Z$ = "N" THEN 270: REM REBOU
    CLE
890 PRINT D$"RUN CATGEN"
900 STOP
910 PRINT D$: PRINT "ERREUR ": PEEK (22
    2): " LIGNE ": PEEK (218) + PEEK
    (219) * 256: STOP

```

Programme SPEED SORT

```

1000- 20 E3 DF 8D C0 03 8C C1
1008- 03 20 BE DE 20 E3 DF 8D
1010- CA 03 8C CB 03 A9 00 8D

```

```

1018- C2 03 8D C3 03 AD C1 03
1020- 85 FC AD C0 03 85 FB A0
1028- 02 A2 01 B1 FB 95 08 CA
1030- 88 D0 F8 A9 00 8D C8 03
1038- 8D C9 03 AD CA 03 85 FD
1040- AD CB 03 85 FE AD C1 03

```

```

1048- 85 FA AD C0 03 85 F9 A0
1050- 02 A2 01 B1 F9 95 06 CA
1058- 88 D0 F8 AD C4 03 8D C6
1060- 03 AD C5 03 8D C7 03 A0
1068- 00 B1 06 D1 08 D0 05 C8
1070- C0 1E D0 F5 F0 0A B0 08
1078- EE C8 03 D0 03 EE C9 03
1080- 18 A5 F9 69 03 85 F9 A5
1088- FA 69 00 85 FA A0 02 A2
1090- 01 B1 F9 95 06 CA 88 D0
1098- F8 18 B0 97 CE C6 03 D0
10A0- 03 CE C7 03 AD C7 03 10

```

```

10A8- BE AD C6 03 10 B9 0E C8
10B0- 03 2E C9 03 18 A5 FD 6D
10B8- C8 03 85 FD A5 FE 6D C9
10C0- 03 85 FE A0 00 AD C3 03
10C8- 91 FD C8 AD C2 03 91 FD
10D0- EE C2 03 D0 03 EE C3 03
10D8- AD C5 03 CD C3 03 D0 09
10E0- AD C4 03 CD C2 03 B0 01
10E8- 60 18 A5 FB 69 03 85 FB
10F0- A5 FC 69 00 85 FC A0 02
10F8- A2 01 B1 FB 95 08 CA 88
1100- D0 F8 38 B0 95

```

```

100 REM *****
110 REM * CATALOGUE.. *
120 REM * PHASE: SAISIE *
130 REM *****
140 REM BUFFER EN '9400
150 HIMEM: 37888
160 ONERR GOTO 2900
170 DIM NAS(100),TP$(100),SE(100),CAS(1
    00)
180 DIM PI(100),AS(100),LG$(100),TA(100
    ),NL$(100)
190 REM NAS=NOM DES FICHIERS
200 REM TP$=TYPE DES FICHIERS
210 REM SE=AD. SECTEUR
220 REM PI=NUM. DE PISTE
230 REM AS=ADR.CHARGEMENT DES PROG. BIN
    AIRES
240 REM LG$=LONGUEUR DES PROG. BINAIRES

250 REM TA=TAILLE DES FICHIERS
260 DIM ST$(100)
270 GOSUB 2910: REM S.P. DE LECTURE '
    RWTS'
280 DEF FN X(A) = (176 + A) * (A - 9 <
    = 0) + (183 + A) * (A - 9 > 0)
290 V20$ = " ": REM
    20 ESPACES
300 D$ = CHR$(4)
310 F$ = "CATALOGUE GENERAL"
320 RD$ = D$ + "READ "
330 WR$ = D$ + "WRITE "
340 OP$ = D$ + "OPEN "
350 CL$ = D$ + "CLOSE "
360 DL$ = D$ + "DELETE"
370 REM LECTURE DE LA DATE =>DA$
380 FOR I = 1 TO 8
390 DA$ = DA$ + CHR$(PEEK(800 + I))
400 NEXT I
405 DK = PEEK(810)
410 TT = PEEK(809) - 2: GOTO 1920
420 REM BOUCLE DE CATALOG
430 VTAB 4: HTAB 10: VTAB 11: FLASH : P
    RINT "LECTURE";: NORMAL : PRINT "
    DU CATALOGUE"
440 SE = 15
450 AI = 768: POKE AI + 15,SE: POKE AI +
    14,17
460 CALL AI

```

```

470 ZT = 37902: REM ZT='9400+14
480 NAS = ""
490 FOR I = 1 TO 7
500 J1 = ZT + 35 * (I - 1)
510 FOR J = J1 TO J1 + 20: REM POUR 20
    CAR. MAX
520 AC = PEEK(J)
530 IF J - J1 = 0 AND AC = 0 THEN 750:
    REM TEST DE FIN DE CATALOGUE
540 IF PEEK(J - 3) = 255 AND J - J1 =
    0 THEN 720: REM TEST DE FICHIER
    'DELETED'
550 IF AC = 34 OR AC = 128 + 34 THEN AC
    = 39: REM REMPLACE " PAR '
560 NAS = NAS + CHR$(AC - 128 * (AC >
    128)): REM CONSTITUTION DU NO
    M DE FICHIER
570 NEXT J
580 NF = NF + 1
590 PI(NF) = PEEK(J1 - 3):SE(NF) = PE
    EK(J1 - 2)
600 IF LEN(NAS) > 20 THEN NAS = LEFT
    $(NAS,20)
610 NAS(NF) = NAS
620 A1 = PEEK(J1 - 1)
630 TA(NF) = PEEK(J1 + 30)
640 NAS = "":TP$ = ""
650 IF A1 > = 128 THEN A1 = A1 - 128:T
    P$ = "": REM TEST DE FICHIER 'L
    OCKED'
660 IF A1 = 0 THEN TP$ = TP$ + "T"
670 IF A1 = 1 THEN TP$ = TP$ + "I"
680 IF A1 = 2 THEN TP$ = TP$ + "A"
690 IF A1 = 4 THEN TP$ = TP$ + "B"
700 IF A1 = 16 THEN TP$ = TP$ + "R"
710 TP$(NF) = TP$
720 NEXT I
730 SE = SE - 1: POKE AI + 15,SE
740 GOTO 460
750 REM CALCUL DE ADR ET LONG DES FICH
    IERS BINAIRES
760 FOR I = 1 TO NF
770 IF TP$(I) = " B" OR TP$(I) = "*B" T
    HEN GOSUB 820:AS(I) = AR$:LG$(I)
    = LG$: GOTO 790
780 AS(I) = "....":LG$(I) = "...."
790 REM CALCUL DE AD$ ET LG$
800 NEXT I
810 RETURN
820 POKE AI + 14,PI(I): POKE AI + 15,SE
    (I)
830 ZT = 37888

```

```

840 CALL AI
850 POKE AI + 15, PEEK (ZT + 13)
860 CALL AI
870 AA = PEEK (ZT + 1)
880 A = INT (AA / 16): B = AA - A * 16
890 AS = CHR$ (FN X(A)) + CHR$ (FN X
(B))
900 AA = PEEK (ZT)
910 A = INT (AA / 16): B = AA - A * 16
920 ARS = AS + CHR$ (FN X(A)) + CHR$
(FN X(B))
930 AA = PEEK (ZT + 3)
940 A = INT (AA / 16): B = AA - A * 16
950 AS = CHR$ (FN X(A)) + CHR$ (FN X
(B))
960 AA = PEEK (ZT + 2)
970 A = INT (AA / 16): B = AA - A * 16
980 LGS = AS + CHR$ (FN X(A)) + CHR$
(FN X(B))
990 RETURN
1000 REM AFFICHAGE
1010 HOME
1020 INVERSE : PRINT "DISQUETTE : ";
NORMAL : HTAB 16: PRINT LEFT$ (N
DS,4);
1030 HTAB 30: INVERSE : PRINT "FACE :";
NORMAL : HTAB 37: PRINT RIGHTS
(ND$,1)
1040 INVERSE : PRINT "LABEL : ";
HTAB 16: NORMAL : PRINT LBS
1050 INVERSE : PRINT "NB. FICHIERS:";
NORMAL : HTAB 16: PRINT NF
1060 GOSUB 1070: GOTO 1180
1070 REM NOM DES RUBRIQUES
1080 VTAB 15: RESTORE
1090 READ Z$: IF Z$ < > "RUB" THEN 109
0
1100 FOR I = 1 TO 16 STEP 2
1110 READ Z$: IF I < 10 THEN PRINT " "
;
1120 PRINT I;"-";Z$;
1130 HTAB 20
1140 READ Z$: IF I < 9 THEN PRINT " ";
1150 PRINT I + 1;"-";Z$
1160 NEXT I
1170 RETURN
1180 POKE 35,14
1190 CV = 12
1210 VTAB 6: CALL - 958
1220 PRINT "FICHIER"
1240 INVERSE
1250 PRINT "NOM " : PRINT "TYPE
"
1260 PRINT "TAILLE " : PRINT "ADRESSE
"
1270 PRINT "LONGUEUR " : PRINT "FICH.LIE
"
1280 PRINT "CLASSE "
1290 NORMAL
1292 FOR I = 1 TO NF
1294 VTAB 6: HTAB CV: PRINT I
1296 FOR J = 1 TO 7: VTAB 6 + J: HTAB C
V: CALL - 868: NEXT J
1300 VTAB 7: HTAB CV: IF TT = 0 THEN P
RINT NAS(I): GOTO 1330
1310 INPUT "": NAS: IF LEN (NAS) = 0 TH
EN VTAB 7: HTAB CV: GOTO 1310
1320 FOR N = LEN (NAS) TO 19: NAS = NAS
+ " ": NEXT N: NAS(I) = NAS
1330 VTAB 8: HTAB CV: IF TT = 0 THEN P

```

```

RINT TP$(I): GOTO 1360
1340 INPUT "": TP$: IF LEN (TP$) = 1 TH
EN TP$(I) = " " + TP$
1350 TS = RIGHTS (TP$,1): IF TS < > "A
" AND TS < > "B" AND TS < > "I"
AND TS < > "R" AND TS < > "T"
THEN 1330
1360 VTAB 9: HTAB CV: IF TT = 0 THEN P
RINT TA(I): GOTO 1400
1370 INPUT "": TAS: IF VAL (TAS) = 0 TH
EN 1360
1380 TA(I) = VAL (TAS)
1390 IF TT = 1 AND TS < > "B" THEN AS(I
) = "....": LGS(I) = "....": GOTO
1460
1400 VTAB 10: HTAB CV: IF TT = 0 THEN
PRINT AS(I): GOTO 1430
1410 INPUT "": AS(I): IF LEN (AS(I)) =
0 THEN AS(I) = "...."
1420 IF LEN (AS(I)) < 4 THEN 1400
1430 VTAB 11: HTAB CV: IF TT = 0 THEN
PRINT LGS(I): GOTO 1460
1440 INPUT "": LGS(I): IF LEN (LGS(I))
= 0 THEN LGS(I) = "...."
1450 IF LEN (LGS(I)) < 4 THEN 1430
1460 IF CG$ = "O" THEN NLS(I) = V20$: CA
$(I) = GH$: VTAB 13: HTAB CV: PRI
NT GH$: GOTO 1630
1470 VTAB 12: HTAB CV: PRINT V20$: VTA
B 12: HTAB CV: INPUT "": NS: NLS =
NS: FOR N = LEN (NS) TO 19: NLS =
NLS + " ": NEXT N
1480 IF LEN (NS) = 0 THEN NLS = V20$:
GOTO 1540
1490 REM TEST EXISTENCE DE FIC. LIE
1500 FOR J = 1 TO NF
1510 IF NLS = NAS(J) THEN 1540
1520 NEXT J
1530 GOTO 1470
1540 NLS(I) = NLS
1550 VTAB 13: HTAB CV: PRINT " " : H
TAB CV: INPUT "": CAS
1560 IF LEN (CAS) = 0 AND LEN (GH$) <
> 0 THEN VTAB 13: HTAB CV: CAS
= GH$: PRINT CAS: GOTO 1600
1570 IF LEN (CAS) < > 0 THEN 1600
1580 IF LEN (CAS) = 0 AND I = 1 THEN 1
550
1590 IF LEN (CAS) = 0 AND I > 1 THEN C
AS = CAS(I - 1): VTAB 13: HTAB CV
: PRINT CAS
1600 CL = VAL (CAS): IF CL < 1 OR CL >
16 THEN 1550
1610 CAS(I) = STR$ (CL): IF CL < 9 THEN
CAS(I) = "0" + CAS(I)
1630 IF CG$ = "O" THEN 1660
1640 VTAB 23: INVERSE : PRINT "OK (O/N
OU 'ESC')?": NORMAL : HTAB 18: P
RINT " " : GET Z$
1650 IF Z$ = "N" THEN 1470
1655 IF Z$ = CHR$ (27) THEN NAS(I) = "
"
1660 NEXT I
1670 POKE 34,2: POKE 35,24
1680 RETURN
1690 REM CONSTITUTION DE L'ARTICLE
1700 REM NO$(4)=NUMERO D'ARTICLE
1710 REM ND$(4)=NUMERO DE DISQUETTE
1720 REM FAS(1)=FACE A/B
1730 REM NAS(20)=NOM DU FICHIER

```



```

1740 REM TP$(2)=TYPE DU FICHIER
1750 REM TA$(3)=TAILLE EN SECTEURS
1760 REM AD$(4)=ADRESSE DE CHARGEMENT
1770 REM LG$(4)=LONGUEUR
1780 REM NL$(20)=FICHIER LIE
1790 REM CA$(2)=CLASSE DU PROG.
1800 REM DA$(8)=DATE
1810 REM LBS(12)=LABEL DISQUETTE
1820 TAS = STR$(TA(I)):L = LEN (TAS):
    IF L = 3 THEN 1900
1830 FOR N = L TO 2:TAS = " " + TAS: NEXT N
1900 ST$ = ND$ + NA$(I) + TP$(I) + TAS +
    AS(I) + LG$(I) + NL$(I) + CA$(I)
    + DA$ + LBS
1910 RETURN
1920 REM MISE A JOUR GLOBALE
1930 PC% = 0:ID = 0
1940 HOME :TS = "SAISIE/MISE A JOUR": H
    TAB 20 - LEN (TS) / 2: INVERSE :
    PRINT TS: NORMAL : PRINT
1950 IF TT = 1 THEN 1970
1960 PRINT "MONTER LA DISQUETTE ";; INV
    ERSE : PRINT "A ANALYSER": NORMAL
    : PRINT "SUR LE DRIVE ";; INVER
    SE : PRINT DK;; NORMAL : PRINT "
    ET TAPER 'RETURN'": GET Z$: PR
    INT Z$
1970 VTAB 5: CALL - 958: PRINT "EST-CE
    UNE NOUVELLE DISQUETTE (";; INVE
    RSE : PRINT "O/N": NORMAL : PRIN
    T ") ? ";; GET Z$: PRINT Z$
1980 IF Z$ < > "O" AND Z$ < > "N" THE
    N 1970
1990 MA = 0 * (Z$ = "O") + 1 * (Z$ = "N"
    )
2000 NF = 0
2010 REM LECTURE DANS ND$=NO DISQUETT
    E-FACE
2020 VTAB 6: INPUT "NUMERO DE LA DISQUE
    TTE ";;ND$
2030 IF LEN (ND$) = 0 OR LEN (ND$) >
    4 THEN 2020
2040 IF LEN (ND$) = 4 THEN 2060
2050 FOR I = 3 TO LEN (ND$) STEP - 1:
    ND$ = "0" + ND$: NEXT I
2060 VTAB 7: HTAB 10: PRINT " FACE ( A
    OU B ) : ";; GET FA$: PRINT FA$
2070 IF FA$ < > "A" AND FA$ < > "B" T
    HEN 2060
2080 ND$ = ND$ + FA$
2090 VTAB 8: INPUT "LABEL DISQUETTE(0..
    12 CAR): ";;LBS
2100 IF LEN (LBS) > 12 THEN 2090
2105 IF LEN (LBS) = 12 THEN 2120
2110 FOR N = LEN (LBS) TO 11:LBS = LBS
    + " ": NEXT N
2120 VTAB 13: INVERSE : PRINT "CLASSE D
    ES FICHIERS": NORMAL : GOSUB 1070
    : VTAB 9: INPUT "CLASSE DES FICHI
    ERS(1..16): ";;GH$: IF LEN (GH$)
    = 0 THEN 2170
2130 GH = VAL (GH$): IF GH < 1 OR GH >
    16 THEN 2120
2140 GH$ = STR$(GH): IF GH < 9 THEN GH
    $ = "0" + GH$
2150 VTAB 10: PRINT "EST-ELLE GLOBALE (
    O/N) ? ";; GET CG$: PRINT CG$
2160 PRINT
2170 IF TT = 1 THEN INPUT "COMBIEN Y-A

```

```

-T-IL DE FICHIERS ? ";;NF$:NF = V
    AL (NF$): IF NF = 0 THEN 2170
2180 ND$(ID) = ND$:ID = ID + 1
2190 IF TT = 0 THEN GOSUB 420: REM LE
    CTURE DU CATALOG
2200 GOSUB 1000: REM AFFICHAGE
2210 REM TRANSFERT DE ST$ DANS ST$(K)
2220 REM PC%=POINTEUR COURANT DANS S
    T$(K)
2230 J = 0:I = 1
2235 IF LEN (NA$(I)) = 0 THEN 2260
2240 GOSUB 1820:J = J + 1
2250 ST$(J + PC%) = ST$
2260 I = I + 1: IF I < = NF THEN 2235
2270 PC% = PC% + J
2280 HTAB 1: VTAB 23: CALL - 868: HTAB
    30: PRINT "TAUX: "PC%": VTAB 2
    3: INVERSE : PRINT "ENCORE (O/N)
    ? ";; NORMAL : PRINT " ";; GET Z$
    : PRINT Z$;
2290 IF Z$ < > "O" AND Z$ < > "N" THE
    N 2280
2300 IF Z$ = "O" THEN 1940
2310 VTAB 23: CALL - 868: HTAB 1: INVE
    RSE : PRINT "ECRITURE SUR DISQUET
    TE (O/N)+RETURN ?": NORMAL : INP
    UT " ";;Z$
2320 IF Z$ < > "O" AND Z$ < > "N" THE
    N 2310
2330 IF Z$ = "O" THEN GOTO 2350: REM
    MISE A JOUR DU FICHIER
2340 GOTO 2830
2350 REM DEBUT DE LA SESSION DE SAISIE
    /MISE A JOUR
2360 HOME : HTAB 10: INVERSE : PRINT "S
    AISIE/MISE A JOUR": NORMAL : PRIN
    T
2370 IF PEEK (810) = 2 THEN 2410
2380 PRINT "MONTER LA DISQUETTE ";; INV
    ERSE : PRINT "PROGRAMME": NORMAL
    : PRINT "SUR LE DRIVE ";; INVERSE
    : PRINT 1;; NORMAL : PRINT " ET
    TAPER 'RETURN'";
2400 GET Z$: PRINT Z$
2410 IF MA = 0 THEN 2620
2420 REM MARQUAGE POUR EFFACER
2430 PRINT : FLASH : PRINT "MARQUAGE DE
    S FICHIERS A SUPPRIMER": NORMAL
2450 PRINT OP$F$
2460 PRINT RD$F$;"R": INPUT NO$:N1% =
    NO$
2470 INPUT LO$
2480 IF NO$ = 1 THEN 2610
2490 PRINT CL$F$
2500 PRINT OP$F$;"L":LO$
2510 FOR IN = 1 TO N1% - 1
2520 PRINT RD$F$;"R":IN
2530 INPUT S1$:NND$ = MID$(S1$,6,5):
    REM =N.DISK
2540 FOR IK = 0 TO ID
2550 IF ND$(IK) < > NND$ THEN 2590
2560 ST$ = CHR$(34) + LEFT$(S1$,4) +
    " " + RIGHT$(S1$,LO$ - 7)
2570 PRINT WR$F$;"R":IN
2580 PRINT ST$
2590 NEXT IK
2600 NEXT IN
2610 PRINT CL$F$
2620 REM MISE A JOUR DU FICHIER
2630 VTAB 10: FLASH : PRINT "ECRITURE";

```

```

: NORMAL : PRINT " DANS " ;: INV
ERSE : PRINT F$: NORMAL
2640 REM OUVERTURE DE F$
2650 PRINT OPSFS
2660 PRINT RDFS$;" ,RO": INPUT NO%:N1% =
NO%
2670 INPUT LO%
2680 PRINT CLSFS
2690 PRINT OPSFS;" ,L":LO%
2700 FOR I = 1 TO PC%
2710 NOS = STR$ (NO%):L = LEN (NOS): F
OR N = L TO 3:NOS = "0" + NOS: NE
XT N
2720 ST$ = CHR$ (34) + NOS + " " + ST$(
I):ST$(I) = ""
2730 PRINT WRSFS;" ,R":NO%
2740 PRINT ST$
2750 NO% = NO% + 1
2760 NEXT I
2770 PRINT CLSFS
2780 PRINT OPSFS
2790 PRINT WRSFS;" ,RO"
2800 PRINT NO%: PRINT LO%
2810 PRINT CLSFS
2820 PRINT : PRINT "IL Y A " :NO% - 1;"
ARTICLES."
2830 VTAB 23: CALL - 958: INVERSE : PR
INT "RETOUR AU MENU (O/N) ?": NO
RMAL : PRINT " " : GET Z$: PRINT
Z$:
2840 IF Z$ = "N" THEN 1920: REM ON REB

```

```

OUCLE
2850 VTAB 4: CALL - 958
2860 IF PEEK (810) = 1 THEN PRINT : P
RINT "MONTER LA DISQUETTE " ;: INV
ERSE : PRINT "PROGRAMME": NORMAL
: PRINT "SUR LE DRIVE " ;: INVERSE
: PRINT 1: NORMAL : PRINT " ET
TAPER 'RETURN' " ;: GET Z$: PRINT Z
$
2870 CCS = "CATGEN"
2880 PRINT D$:"RUN"CC$: STOP
2900 PRINT D$: PRINT "ERREUR " : PEEK (2
22):" LIGNE " : PEEK (218) + PEEK
(219) * 256: STOP
2910 REM INIT. SEQUENCE 'RWTS'
2920 RESTORE
2930 FOR I = 0 TO 30: READ ML: POKE 768
+ I,ML: NEXT I
2940 DATA 169,3,160,10,32,217,3,96,0,0,
1,96,1,0,17,15,27,3,0,148,0,0,1,0
,0,96,1,0,1,239,216
2950 POKE 780, PEEK (810)
2960 RETURN
2970 DATA RUB
2980 DATA REFERENCES,SYSTEME,UTILITAIRE
,PROCEDURE,JEUX REFLEXION,JEUX AD
RESSE
2990 DATA GRAPHIQUE 1,GRAPH 2,EDUCATIF
,GEOGRAPHIE,MATHS 1,PHYSIQUE,MUSI
QUE
3000 DATA DONNEES,GESTION,DIVERS

```

GAGNEZ



EN ACHETANT

VOS LOGICIELS PAR CORRESPONDANCE!

Extrait de nos tarifs :

	FF TTC
MULTIPLAN	1750
CARTE APPLE-TELL	5550
JANE	1420
EPISTOLE //c (souris)	1560
PFS: WRITE	1140
PFS: REPORT	1140
PFS: FILE	1140
PFS: GRAPH	1230
VISICALC AVANCE //e	2800
CX BASE 200 + CX TEXTE	3190
DBASE II (CP/M)	6070
FRIDAY! (CP/M)	2660

* Plus ou moins selon le produit.

	FF TTC
PIE:WRITER	1950
THINKTANK	1760
OMNIS 1	1850
OMNIS 2	3320
OMNIS 3	6550
DECISIONNEL GRAPHIQUE	1850
COMPTABILITE CYRUS	4270
EDI-LOGO	1190
SARGON III	470
FLIGHT SIMULATOR II	550
SORCELLERIE 1	520
SORCELLERIE 2	400

Tous nos envois sont en recommandé. Les frais de port sont inclus. Envoyez votre commande accompagnée d'un chèque bancaire ou postal à l'ordre de INFOSOFT, en précisant votre configuration et votre type de matériel. Tous ces logiciels sont destinés à l'APPLE II. Pour tout renseignement ou pour recevoir nos tarifs complets, écrivez-nous à l'adresse ci-contre.

INFOSOFT
BP 516
75066 Paris Cedex 02

Koala Pad est un logiciel éditeur de dessins et graphiques, vendu d'origine avec une disquette et une tablette sensitive, au prix de 1600 FF TTC. Ce qui le distingue des autres programmes de création graphique, c'est avant tout sa simplicité de mise en oeuvre. Il comporte un programme extrêmement rapide contenant toutes les commandes graphiques haute résolution (HGR). Il requiert un Apple II ou //e avec 48 Ko de mémoire centrale et un lecteur de disquettes. Néanmoins, pour en faire un outil de travail, il est préférable de disposer de deux lecteurs de disquettes. Le copyright est de "Island Graphics" et la diffusion assurée par "Technology Corporation".

La tablette sensitive se connecte sur la sortie manette de jeux de l'Apple. La version que j'ai testée était modifiée de façon à utiliser le joystick //e, rendant le programme plus performant et plus souple à l'emploi. L'essai a été fait sur un Apple //e, mais l'utilisation est possible sur tout Apple II.

L'abord du logiciel est assez aisé, et l'on acquiert très vite la maîtrise de l'ensemble des fonctions. On peut diviser Koala en deux parties distinctes : les fonctions utilitaires et les fonctions graphiques.

Fonctions utilitaires

L'appel de cette rubrique affiche un menu global à l'écran :

C)atalog
U)tilitaires étendus
L)oad
S)ave

Trois de ces fonctions sont bien connues de nos lecteurs. La fonction "Utilitaires étendus", quant à elle, affiche un sous-menu :

L)ock
U)nlock
V)erify
R)ename
D)elete
F)ormat data disk

Cette dernière fonction permet de formater une disquette sans le DOS (pour gagner de la place), avec au maximum une disquette "gonflée" de 528 sec. pour stocker vos chefs-d'oeuvres.

En somme, le concepteur n'a pas oublié les fonctions élémentaires du DOS, qui font parfois défaut sur des programmes d'un niveau supérieur.

Fonctions graphiques

Lorsque le programme est chargé en mémoire, l'écran affiche un menu très complet sous forme d'un tableau. Aucune connaissance en informatique n'est requise pour utiliser le programme.

Il y a 15 instructions de base (qui rappellent beaucoup les fonctions graphiques de LISA par leur simplicité) et une palette de 16 couleurs. Il existe en outre une fonction "CURSOR", qui permet de modifier la structure du curseur, en forme de point ou de croix, ce qui facilitera le pointage sur la page HGR. En dernier, nous trouvons la fonction "BRUSH SET", qui donne la possibilité de dessiner comme si l'on avait un pinceau à la main.

Détail des fonctions graphiques

DRAW : dessin en continu

POINT : dessin de points.

LINE : dessin de traits. On pointe l'origine puis, en se déplaçant sur l'écran à l'aide du joystick, la fin du trait. Celui-ci est alors obtenu en appuyant sur le bouton Q.

LINES : semblable à la fonction "LINE", avec la différence que l'origine de chaque trait est la fin du trait précédent.

RAYS : tracé de rayons à partir d'un centre. Cela aboutit à de beaux effets spéciaux : on pointe l'origine, puis l'on trace des traits partant de cette origine, avec des longueurs réglables.

FILL : remplit un dessin (éventuellement l'écran entier) avec la couleur choisie. La surface à remplir doit être délimitée par un trait continu; sinon, il risque d'y avoir des surprises (remplissage intempestif de toute une zone connexe).

FRAME : tracé de rectangles. Il suffit de pointer un sommet et d'ajuster la surface (longueur et largeur) avec le manche du joystick; on confirme avec le bouton Q, lorsque la figure obtenue est adéquate.

BOX : équivaut à "FRAME" puis "FILL". Création d'un rectangle plein.

CIRCLE : tracé de cercles. On pointe le centre du cercle, puis on délimite le rayon avec le joystick.

DISC : équivaut à "CIRCLE" puis "FILL". Création d'un disque plein.

ERASE : annule le dessin en cours, en remplissant l'écran de la couleur choisie.

STORAGE : donne accès aux fonctions utilitaires décrites plus haut.

NORMAL : mode d'utilisation normale, c'est-à-dire à l'échelle 1.

MAGNIFY : mode loupe. Ce mode d'utilisation agrandit le dessin afin de faciliter la correction des défauts, comme dans le fameux Pinball Construction Set, ou comme Micro Painter. Très utile.

Utilisation de la couleur

Le choix des couleurs se divise en deux groupes de couleurs : le "color

Koala pad

Marcel Cottini

set 1" et le "color set 2". Les habitués du graphisme couleur sauront de quoi il retourne. Pour les non-initiés, voici un petit rappel.

L'Apple utilise pour son graphisme deux pages en haute résolution, la page 1 (\$2000-\$3FF7) et la page 2 (\$4000 à \$5FF7). Chaque série de 7 points sur une même ligne de l'écran correspond à un octet en mémoire. Un point est allumé lorsque le bit correspondant est à 1. Une ligne est donc formée de 40 séries de 7 points (280 points par ligne en HGR). Voir à ce sujet "Graphiques : de l'ITT 2020 à l'Apple" et "Les adresses du graphique", publiés dans le Pom's 1.

Le bit de poids fort de chaque octet (bit numéro 7) n'est pas visualisé, mais il conditionne en partie la couleur des 7 points liés à l'octet dont il fait partie. Ainsi, si le bit 7 est à 0, tout point isolé correspondant à cet octet et situé en colonne paire apparaît violet; en colonne impaire, il apparaît vert. Si le bit 7 est à 1, les points isolés en colonne paire apparaissent bleus, ceux en colonne impaire apparaissent rouges. Si deux points ou plus sont côte à côte sur une même ligne, ils apparaissent blancs même s'ils ne dépendent pas du même octet. Pour de plus amples précisions, vous pouvez aussi vous reporter au manuel de référence de l'Apple II.

Ceci nous amène à déborder un peu du sujet, mais explique l'existence de deux gammes de couleurs. Une précaution s'imposera donc lorsque vous travaillerez sur un graphique : c'est de toujours utiliser une couleur appartenant au même "SET" que la couleur du fond. Vous éviterez ainsi les bavures et mélanges de couleur indésirables.

Fonction Menu/Page graphique

Lorsque le programme est lancé, un menu sous forme de tableau s'affiche à l'écran, représentant les symboles et repères des différentes fonctions à votre disposition. Pour sélectionner une fonction donnée, il suffit de placer le curseur (matérialisé par un point) à l'aide du joystick sur le symbole de votre choix.

Exemple : pour dessiner un cercle, vous choisissez successivement la couleur de fond, la finesse du trait, la couleur du trait et enfin la fonction "CIRCLE".

Une fois votre choix arrêté, vous pointez le curseur au bas de l'écran

Micro-informations

Jean-Michel Gourévitch

Où va donc Apple ? Un Macintosh pas tout à fait assez puissant pour affronter l'IBM PC sur le terrain des applications professionnelles, un Apple //c pas tout à fait compatible avec ses congénères de la gamme Apple II, l'accélération du mouvement chez le constructeur à la pomme multicolore donne le vertige aux aficionados. Où va Apple ? Pour le Macintosh, la réponse est : vers le "Big Mac", un Mac survitaminé (512K contre 128) avec un second drive et dont l'apparition (en raison du coût des chips 256K) ne se fera sans doute que l'an prochain.

Où va Apple ? Pour la gamme Apple II, quelques éléments de réponse se trouvent dans une passionnante interview accordée par Steve Wozniak à la revue "In Cider" dans son numéro de juillet. "Woz the wiz" (Woz le magicien), comme l'ont surnommé les Américains, réaffirme notamment la fidélité d'Apple à la compatibilité. Tout en assurant que des logiciels utilisant la double haute résolution graphique (donc incompatibles avec certains AII+) sortiront dès la fin de l'année. Pour la souris, pas de problème : elle est appelée à devenir un standard sur l'Apple II (même si les logiciels d'exploitation style Macintosh mettront encore quelques années à voir le jour).

Le plus intéressant, c'est à coup sûr ce qu'annonce Steve Wozniak sur les développements de l'Apple //. Pas question d'aller vers une intégration du MS-DOS (le standard des compatibles IBM), on s'en serait douté. En revanche, l'avenir de l'Apple // passe par le processeur 65816. Un 16 bits compatible patte pour patte avec le 6502 des Apple //, qu'il émule au repos (on peut sans difficultés extraire l'un pour le remplacer par l'autre). Vrai 16 bits ou pas ? Pour Wozniak, ceux qui posent cette question sont "à côté de la plaque". L'important, explique-t-il, c'est que le 65816 dispose de 24 bits adressables. L'intérêt ? "On est capable, alors, d'adresser 16 Megabytes au lieu de 64K". Et ça, c'est tout à fait passionnant, car rien n'interdit plus alors d'ajouter à l'Apple des cartes d'extension mémoire équipées de chips de 256K, d'écrire et d'utiliser des logiciels super-intégrés (genre Open Access ou Lotus Symphony), gros consommateurs de mémoire, qui font le triomphe de l'IBM PC.

De quoi rêver, non ? Seulement, nous replongeons alors dans les

affaires de l'incompatibilité. Si doper un Apple II+ ou //e est possible, réaliser la même opération sur le //c paraît plus problématique (les chips sont soudés sur la carte mère et non enfichés comme sur les autres Apple //). Conclusion : gare au //c ! Il n'est pas compatible à 100 % avec ses aînés, et réciproquement : ainsi, si Jane tourne indifféremment sur n'importe quel Apple, Epistole version //c-souris produit d'étranges effets sur un //e muni d'une souris. On risque donc à terme de se retrouver avec les 16.000 programmes existants, avec une gamme écrite pour le //c et ne fonctionnant que sur cette machine, et avec des logiciels "pros" pour Apple // 16 bits et réservés à ce dernier. Mais, après tout, il existe déjà des logiciels pour Apple nécessitant une carte spéciale (les produits CP/M).

En attendant les Apple de demain, voici de quoi occuper les mémoires de ceux d'aujourd'hui.

Une souris pour des pommes

Le jeu de mot est douteux. Pas la souris, qui vient d'arriver pour l'Apple // : une carte d'extension (une de plus), un manuel, une souris et une disquette, c'est ce que propose Apple pour quelques 1200 F. La souris, c'est celle de Macintosh. La carte possède son propre microprocesseur (qui rend les opérations de la souris plus rapides qu'avec le //c). Le manuel est parfait et comporte même des routines pour utiliser la bête en Basic. Quant à la disquette, elle offre "Mouse Paint", un logiciel de dessin qui reprend certaines des fonctions de MacPaint (pour Macintosh). Le trait est certes moins fin que sur le 32 bits. Seulement, là, on a la couleur.

Plus traditionnelles, les cartes musicales d'ALF pour Apple // à trois voies MC1 (169 dollars) ou à neuf voies MC16 (179 dollars). On peut changer le tempo ou jouer plusieurs airs en séquence. Ou encore ce buffer de clavier Quentin AP 810, qui permet aux possesseurs d'Apple //e de taper sur les touches à leur rythme sans se préoccuper de savoir si la machine suit.

Donner de la voix

Avec l'Introvoice //, on peut désormais contrôler des logiciels en par-

lant. Une carte, à enficher (qui contient un analyseur de spectre audio, 16K de RAM et 8K de ROM), un micro et un logiciel. Le constructeur garantit la reconnaissance d'un vocabulaire de 160 mots maximum avec une précision de 98%. Même si ces appareils ont quelques difficultés à s'adapter à un vocabulaire en français, ce n'est pas mal (995 dollars pour l'Apple //e).

Espionner la police

Ou encore écouter les communications des radio téléphones de voitures, des taxis ou des dépanneurs, avec l'aide de l'Apple // et du scanner Bearcat CP 2100 d'Electra. Deux cents canaux programmables, dont les caractéristiques, les fréquences et les noms peuvent être visualisés sur le moniteur ou mis en mémoire sur disquette. La radio avec son haut parleur et son antenne se relie par câble à une interface RS 232. Ensuite, l'Apple prend le contrôle et on pilote au clavier le niveau sonore, le réglage des fréquences, etc. (500 dollars).

Cocorico

Voici un Modem français. Et pour moins de 1500 francs, en voie d'homologation de surcroît, c'est celui d'In Electronic (le créateur des célèbres Modulo-Phones). Il sera présenté à la rentrée au SICOB.

L'Apple art

Voici de quoi exploiter les possibilités graphiques des Apple //. Flying Colors permet de réaliser des dessins assistés par ordinateur avec le joystick, et aussi de présenter des images sur l'écran (manuellement ou automatiquement) selon un ordre programmable. B/ Graph, de Commodore, tourne aussi sur Apple et offre un grand choix de graphiques et un excellent manuel de statistiques (100 dollars). Une nouvelle version de Graphics Department de Sensible Software permet de faire une présentation graphique (courbes, graphiques, camemberts, le tout agrémenté de lettres diverses et en séquence, comme avec un projecteur de diapositives) et surtout de la réaliser avec la souris (125 dollars). A signaler, pour ceux que les graphiques intéressent, une étude exhaustive sur les lo-

giciels réalisée par Creative Computing dans son numéro de juillet.

Printshop de Broderbund Software est un logiciel tout à fait étonnant permettant de concevoir, dessiner et imprimer des cartes de vœux, des calicots etc. On choisit ses caractères, le dessin des bordures, les images et les symboles. Le programme est livré avec un choix de papiers de couleurs pour imprimantes à picots et d'autres gadgets (50 dollars). Le SSD (Super Screen Dump) de Schwartz Computer Consulting se compose d'une carte d'interface avec un bouton de recopie écran, d'un manuel et d'un logiciel permettant d'imprimer tous les textes et graphiques sur Imagewriter, DMP ou Epson sans sortir du programme en cours pour un prix défiant toute concurrence (41 dollars). Enfin, après Visicalc, on peut désormais imprimer Multiplan en travers (comme sur le Macintosh) ainsi d'ailleurs que n'importe quel autre programme grâce à Sideways de Funk Software (60 dollars).

Du côté des tableurs, Microsoft sort en septembre la version 1.07 de Multiplan : compatible //c, capable d'utiliser les 128K du //e et du //c, et enrichie de fonctions nouvelles, en particulier le taux de rendement interne que seul Visicalc Advanced Version offrait à ce jour sur Apple. On trouvera aussi les fonctions financières valeur future (FV), nombre de périodes (NPER), mensualités (PMT) et calcul de taux (RATE), cette dernière n'existant pas sur le V.A.V. Les possesseurs de Multiplan version 1.06 pourront se mettre à jour à un prix d'ami (pas plus de 300 francs) grâce à un échange standard que Microsoft compte organiser.

Côté boulot

Typing tutor III de Kriya Systems est un extraordinaire programme permettant d'apprendre la dactylographie. Le programme mesure constamment le temps de réponse de l'élève, et ajuste ses leçons en conséquence. Si ce n'est pas de l'intelligence artificielle, ça y ressemble beaucoup. Pour seulement 50 dollars. Seul défaut, ce programme américain est bien sûr conçu pour un clavier QWERTY. Mais, après tout, apprendre à s'en servir n'est pas forcément du temps perdu. Tick Tack de Longman permet de correspondre dans une langue sans en comprendre un mot. Son génie : des phrases standard, pouvant être traduites automatiquement en anglais, français, allemand ou espagnol. Les lettres peuvent ensuite être éditées grâce à n'importe quel traitement de texte (150 livres en Grande Bretagne).

Pour le Mac et le //c, guère de nouveau

A signaler seulement des disques durs Tecmar Mac Drive de 10 Megabytes fixe ou de 5 Megabytes amovibles pour le Macintosh. Avec les Davong et les Symbiotic qui devraient aussi sortir pour le Mac, on aura bientôt l'embarras du choix.

Pour le //c, Catalyst QC10 de Quark. Les prix ne sont pas encore fixés.

Bien sûr, la plupart de ces produits (qui viennent de paraître) ne sont pas encore disponibles en France. Mais, on peut toujours les commander aux Etats-Unis, et certains ne tarderont pas à être importés. En matière de micro-informatique, la futurologie, c'est déjà le présent, et l'actualité le passé...

Quelques compléments

Multi-Phone, commercialisé par Pro Forma vous permet de programmer des séries d'appels téléphoniques, compose pour vous en faisant apparaître la fiche à l'écran... Une carte reliée à une prise gigogne (pour le téléphone) se loge à l'intérieur de votre Apple II + ou //c; pour le prix forfaitaire de 5.900 francs HT, vous aurez la carte, le logiciel en Pascal, une disquette de démonstration et deux disquettes-fichier contenant les fiches des 1000 premières entreprises françaises classées par effectif décroissant. Vous n'y trouverez pas Pom's...

Alliance Research vend \$300 un clavier détachable pour Apple //c, proche dans son dessin du clavier IBM et possédant comme lui 10 touches de fonction.

Le logiciel **TKI Solver** pour l'analyse et la résolution d'équations devient disponible pour l'Apple //e en version 128K, après avoir fait ses preuves sur le PC d'IBM.

Kodak commercialise dès à présent dans le Royaume Uni un lecteur de disquettes 5 pouces 1.4 avec 3,3 mégaoctets non formatés sur 192 pistes par pouce (tpi), capable de lire aussi du 48 et 96 tpi. Quand on met une disquette normale dans le lecteur, son microprocesseur la détecte aussitôt et adapte la vitesse du moteur en conséquence. On n'arrête pas le progrès!

Le célèbre programme de gestion de fichiers **Omnis**, commercialisé en France par Ka, voit sa gamme s'enrichir : l'utilisateur possède maintenant le choix entre Omnis 1, 2 ou 3. Dans Omnis 3, la version la plus complète, il est possible de définir les menus, les séquences de commande et les

messages les accompagnant; on peut aussi, comme d'ailleurs avec Omnis 2, travailler sur 12 fichiers ouverts simultanément (2 pour dBASE II).

Des fiches en plastique aide-mémoire pour les principaux logiciels vous sont proposées par **Barons AG**, réalisées par Kleertex. Chaque fiche entoure le clavier. Les logiciels déjà pris en compte pour l'Apple sont Wordstar/Mailmerge, Visicalc, dBASE II, Apple Writer et Quick File, ce dernier existant seulement (et pour cause) en version //e.

Adresses

Aif - 1315 F Nelson St. - Denver, Co 80215 - USA

Apple Seedrin - av. de l'Océanie, ZA de Courtabœuf - BP 131 - 91944 Les Ulis - Tél (6) 928.01.39

Barons AG - 8 avenue des Grandes Communes - 1213 Petit Lancy - Genève - SUISSE - Tél (022) 93.47.47.

Broderbund Software - 17 Paul Dr. - San Rafael, CA 94903 - USA

Commodore Business Machines - 1200 Wilson Dr. - West Chester, PA 19380 - USA

Creative Computing (vendu chez SIVEA) - 3460 Wilshire Bd. - Los Angeles, CA 90010 - USA

Electra Company - 300 East County Line Road - Cumberland, IN 46229 - USA

Funk Software - PO Box 1290 Cambridge, MA 02238 - USA

In Cider (en vente à Paris au Multistore Hachette et dans les boutiques Sivea) - 80 Pine St - Peterborough, NH 03458 - USA

In Electronic - 3 Bd. Ney, 75018 Paris - Tél 238.80.85

Kriya Syst - Simon & Schuster Electronic Publ. Div. - 1230 Ave. of the Americas - New York NY 10020 - USA

Longman Group - Longman House - Bunt Mill - Harlow Essex CM 202 JE - Royaume Uni

Microsoft - 519 local Québec - 91946 Les Ulis Cedex - Tél 446.61.36

Pro Forma - 14 rue Martel - 75010 Paris - Tél 523.25.05.

Quark - 2525 West Evans - Suite 220 - Denver, CO 80219 - USA

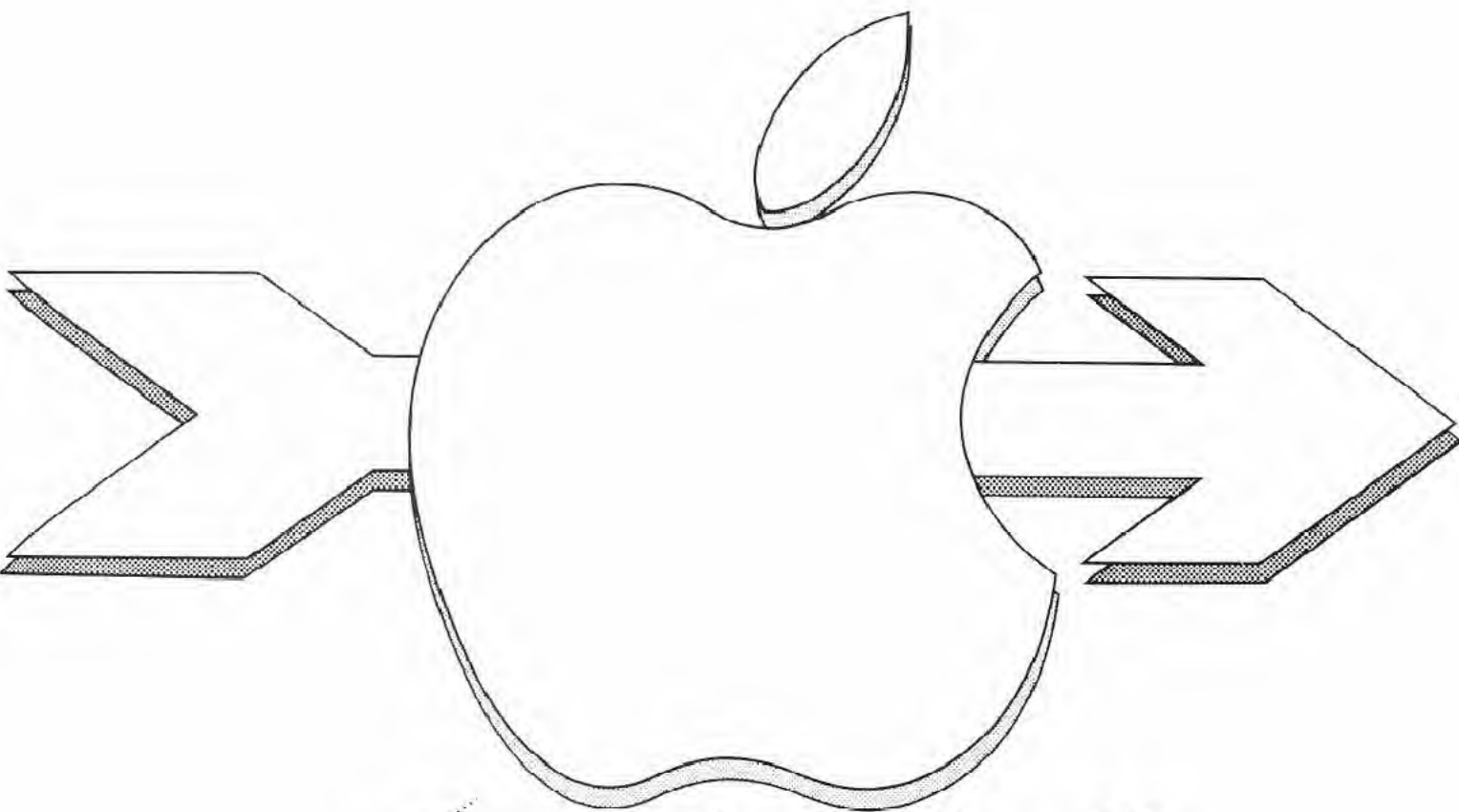
Quentin Corp. - 9207 Eton Ave. - Chatsworth, CA 91311 - USA

Schwartz Computer Consulting - 3868 Cedar Hill Road, Victoria BC - Canada V8P3Z6

Sensible Software - 24011 Seneca Oak Park, MI 48237 - USA

Tecmar Inc. - 6225 Cochran Rd. - Solon (Cleveland) - Ohio 44139 - USA

Voice Machine Communications - 1000 South Grand Ave. - Santa Ana, CA 92725 - USA



apple-tell

un modem qui fait, d'un simple Apple, un Minitel intelligent

Apple-tell comprend :

- Une carte Modem incluant un décodeur Teletel.
- Un logiciel d'Emulation de Terminal Minitel enrichi de trois éblouissantes fonctions (celles qui faisaient le plus défaut jusqu'à présent sur votre Minitel).

IMPRESSION : l'imprimante de votre Apple est exploitée pour sortir les copies-papier dont vous avez besoin lorsque vous consultez un serveur.

STOCKAGE : les disquettes de votre Apple sont utilisées pour enregistrer les pages dont la consultation vous est nécessaire.

- au format Teletel (c'est-à-dire telles que vous les avez reçues)
- en mode Texte pur (ASCII) pour exploitation locale ultérieure.

AUTOMATISME : l'intelligence de votre Apple est mobilisée pour accomplir l'interrogation automatique du serveur que vous lui avez désigné (appel téléphonique, orientation TRANSPAC, identification, choix successifs), enregistrer sur papier et/ou sur disque les données consultées puis pour traiter celles-ci, en les incorporant dans votre application.

(Les procédures d'interrogation sont créées par l'utilisateur, sans aucun langage de programmation, grâce au mode d'apprentissage d'Apple-Tell.)

Événement du dernier SICOB, salué par toute la presse, consacré Pomme d'Or 1983 par le jury Apple, le modem Apple-Tell marque une mutation décisive dans l'évolution des techniques vidéotex en environnement professionnel.

- point d'arrêt à la prolifération des matériels sur votre bureau (effet "mini-Sicob").

- Utilisation optimale des ressources dont vous disposez déjà (disques, imprimante, logiciels, etc..)
- Utilisation possible en mode Terminal autant qu'en mode Serveur (jusqu'à quatre portes).
- Enfin (et c'est sans doute le point le plus important) JONCTION entre le monde extérieur et les outils standards de votre Apple : l'incorporation des données dans Apple-writer, Visicalc, Multiplan, PFS, Quick-File, etc., et même dans vos applications personnelles (comptabilité, suivi de commandes, fichier...) devient possible.

CARACTERISTIQUES GENERALES :

- Modem multimodes :
 - 1 200/75, (full-duplex),
 - 1 200 (half-duplex),
 - 600 (half-duplex), 300 (full-duplex),
 - standards CCITT et BELL(cette caractéristique unique rend accessibles les serveurs nord-américains, y compris par réseau téléphonique commuté).

Sorties : video

composé (N & B) et Pétrel

- compatible Apple 2, Apple 2+, Apple 2e (48 K, une disquette)

- Enfilable dans n'importe quel slot libre de votre Apple 2

- Transparence totale vis à vis du système.

HELLO
Informatique
1, rue de Metz
75010 PARIS
Tél. : (1) 523 30 34
Telex : FLASH 210500 F

HELLO Informatique 1, rue de Metz 75010 PARIS Tél. : (1) 523 30 34

☐ Souhaite recevoir une documentation sur le système Apple-Tell
☐ Commande ☐ système(s) Apple-Tell au prix de F. 6997 T.T.C.
règlement ci joint par ☐ chèque bancaire ☐ CCP

IMPRIMANTES **Canon**



Une gamme complète Des technologies de pointe

- Couleur à jet d'encre
- Matricielle
- Laser
- Thermique
- Transfert thermique

MATÉRIEL DISPONIBLE CHEZ LE DISTRIBUTEUR AGRÉÉ

ASAP

3, Avenue des Trois Peuples - 78180 MONTIGNY-LE-BRETONNEUX

Tél. (3) 043.82.33 - Télex 698887 F

Un INPUT généralisé miniature

Jacques Duma

Une des contraintes les plus énervantes de l'Applesoft est peut-être le manque de réalisme de sa routine d'Input. Celle-ci refuse obstinément les virgules et deux-points; et les caractères blancs précédant une chaîne sont purement et simplement oubliés.

Encore une routine d'Input, allez-vous dire... Soit, mais celle que je vous présente aujourd'hui tient en deux lignes Applesoft. Qui fait mieux ?

Utilisation de la routine

La routine tient dans les lignes 100 et 110, et peut être insérée (éven-

tuellement avec d'autres numéros d'instructions) dans tout programme Applesoft. Le petit programme en Basic des lignes 10 à 50 ne sert qu'à illustrer le fonctionnement de cette routine.

Explication du fonctionnement

L'appel en 54572 (\$D52C) utilise la routine d'entrée du moniteur (qui, lui, accepte tout ce qu'on lui donne) et remplit le buffer d'entrée à l'adresse 512 jusqu'à 767 (\$200-\$2FF). Il ne reste plus qu'à transférer la saisie vers une variable chaîne Applesoft : à partir du début du buffer, le premier octet nul indique la fin de

la saisie. Une fois celui-ci trouvé, on obtient l'adresse de la variable par l'expression : $VAL(IN\$) + PEEK(131) + 256 * PEEK(132)$.

En effet, les octets 131 et 132 contiennent en permanence l'adresse du descripteur de la dernière variable utilisée, en l'occurrence IN\$. Il ne reste plus qu'à mettre à jour le descripteur pour qu'il pointe vers la zone du buffer contenant la saisie, puis à copier la chaîne vers une zone mémoire protégée par l'expression $IN\$ = MID\$(IN\$,1)$, apparemment inutile mais qui force la création d'une nouvelle chaîne. Et voilà ! ■

Lecture/Ecriture rapide de tableaux

Jacques Duma

Il est souvent utile de pouvoir sauvegarder un tableau Applesoft de variables numériques, par exemple pour une utilisation conjointe par plusieurs programmes.

Malheureusement cette fonction, disponible avec l'utilisation des cassettes, a disparu avec le DOS 3.3 (mais réapparu sous ProDOS, c'est une autre histoire). Des programmes en assembleur ont parfois été proposés pour combler cette lacune, mais pourquoi faire compliqué quand on peut faire simple ? La routine que je vous propose ici tient en deux lignes de bon vieux Basic : une ligne pour la sauvegarde du tableau, une ligne pour son chargement.

Utilisation

Le programme en Basic de 10 à 210

est un simple exemple d'utilisation de la routine de sauvegarde (ligne 510) et de la routine de lecture (ligne 610). Encore une fois, vous pouvez intégrer ces deux lignes dans tout programme Basic, quitte à leur donner n'importe quel numéro d'instruction.

Fonctionnement

Comme pour l'INPUT généralisé miniature, cette routine utilise avec bonheur les adresses 131 et 132 qui contiennent le vecteur vers le descripteur de la dernière variable référencée, que ce soit une chaîne de caractères ou un tableau de variables numériques. Il ne reste plus qu'à calculer la taille du tableau avec la formule $L = 5 * (N+1) * (P+1)$, où N

et P sont les dimensions, et le tour est joué !

Remarques

Pour un tableau de dimension 3, par exemple, remplacez la formule par $5 * (N+1) * (P+1) * (Q+1)$, où N, P et Q sont les dimensions.

Pour un tableau de nombres entiers, remplacez le 5 de la formule par un 2 puisqu'un entier est stocké sur deux octets seulement.

Enfin, il n'est pas possible d'utiliser un tableau de chaînes de caractères, celles-ci étant dispersées un peu partout dans la mémoire; mais rien ne vous interdit de le convertir en tableau d'entiers par la fonction $ASC(MID\$(T\$(I,J),L,1))$...

```
1 REM ** NIBBLE ARRAY **
10 N = 8:P = 9: REM DIMENSIONS DU TABLEAU
20 DIM T(N,P):T$ = "NIBBLE.TABLEAU"
30 REM ** T(I,J)=1+I*J/10 **
40 FOR I = 0 TO N: FOR J = 0 TO P:T(I,J) = 1 + I + J / 10: NEXT J: NEXT I
50 TEXT : HOME
60 PRINT "AFFICHAGE DES VALEURS": PRINT : GOSUB 200
65 PRINT "SAUVEGARDE DU TABLEAU": GOSUB 500
70 PRINT "EFFACEMENT DU TABLEAU": GOSUB 100
75 PRINT "AFFICHAGE DES VALEURS": PRINT : GOSUB 200
80 PRINT "CHARGEMENT DU TABLEAU": GOSUB 600
85 PRINT "AFFICHAGE DES VALEURS": PRINT : GOSUB 200
90 PRINT : PRINT CHR$(4)"DELETE":T$: END
99 REM ** REMISE A ZERO **
100 FOR I = 0 TO N: FOR J = 0 TO P:T(I,J) = 0: NEXT J: NEXT I: RETURN
199 REM ** AFFICHAGE DES VALEURS **
200 FOR I = 0 TO N: FOR J = 0 TO P: PRINT T(I,J) " "; NEXT J: PRINT : NEXT I: PRINT : RETURN
499 REM ** SAUVEGARDE **
```

```
500 AD = 0 * T(0,0) + PEEK(131) + 256 * PEEK(132): PRINT : PRINT CHR$(4)"BSAVE" T$,A"AD",L*5 * (N+1) * (P+1): RETURN
599 REM ** CHARGEMENT **
600 AD = 0 * T(0,0) + PEEK(131) + 256 * PEEK(132): PRINT : PRINT CHR$(4)"BLOAD" T$,A"AD": RETURN
```

```
1 REM *** NIBBLE INPUT ***
10 TEXT : HOME : PRINT "DONNEZ UNE CHAÎNE DE CARACTÈRES AVEC DES GUILLEMETS, DES VIRGULES OU DES ESPACES:"
20 PRINT : PRINT "-->": GOSUB 100
30 IF IN$ = "" THEN PRINT : PRINT "C'EST FINI": END
40 PRINT : PRINT "=="IN$<==": GOTO 20
100 CALL 54572: FOR B = 512 TO 768: IF PEEK(B) < > 0 THEN NEXT
101 IN$ = "":AD = VAL(IN$) + PEEK(131) + 256 * PEEK(132): POKE AD,B - 512: POKE AD + 1,0: POKE AD + 2,2:IN$ = MID$(IN$,1):B = 768: NEXT : RETURN
```


BONJOUR LES PRIX!!

NOS PRIX SONT F.T.T.C.

Carte langage	400	Speech card	320
Carte 128 k ram	1550	Carte horloge	500
Carte 80 colonnes	700	Joystick	165
Interface série	520	Ventilateur	280
Super série	1000	Contrôleur de drive auto switch 13/16	370
Interface parallèle	380	Microdrive 3"	1900
Grappler + buffer 16 k	1350	Moniteur vert 12"	950
Carte modem	2200	Disquettes 5" 1/4 S.F./S.D. par 1 boîte	140/boîte
Carte Z 80	410	Disquettes 5" 1/4 S.F./D.D. par 1 boîte	175/boîte
Wildcard	400		

AU-DESSUS, NOUS CONSULTER.

Carte bleue et eurocard acceptées
Vente par correspondance: nous consulter.

Computer 3

3, rue Papillon 75009 Paris - Tél.: 523.51.15

(métro Poissonnière) ouverture du lundi au samedi de 10 h à 19 h 30



LA PHOTOCOMPOSITION EN PROLONGEMENT DE LA MICRO-INFORMATIQUE



TRANSMETTEZ-NOUS VOS TEXTES
PAR TÉLÉPHONE

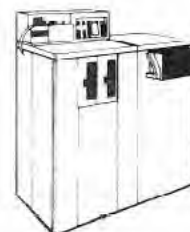
ou

DONNEZ-NOUS VOTRE DISQUETTE



Les textes de vos articles, catalogues, annuaires ou brochures saisis sur votre APPLE sont envoyés directement sur notre photocomposeuse.

Nous vous évitons ainsi, le coût et le temps de la saisie supplémentaire que nécessite le traitement traditionnel de la photocomposition avant l'impression des documents, si vous le désirez nous pouvons également nous charger de l'impression et du brochage.



NOTRE RÉFÉRENCE... LA REVUE POM'S

TELECOMPO 328.18.63

PHOTOCOMPOSITION
BUREAUTIQUE
TRANSMISSION DE DONNÉES

GESTION DE FICHIERS
MATÉRIEL DE
TRAITEMENT DE TEXTES

13 et 15 avenue du Petit Parc
94000 VINCENNES

La touche DEL, sur l'Apple //e, en dehors de faire des damiers qui permettent d'écrire avec des PRINTs le titre Pom's, le sauver sur disquette et le rappeler, a certainement d'autres fonctions. Quel est son code ? Peut-on trouver un exemple simple de son utilisation ?

Jean-Claude Baron -
31530 Lévignac

Quand vous avez un doute sur le code d'une touche, il suffit d'exécuter un petit programme Basic tel que :

```
10 GET A$
20 PRINT ASC(A$)
30 IF ASC(A$)<>13 THEN 10
```

Le programme ci-dessus sort les codes ASCII de toutes les touches et s'arrête dès que l'on appuie sur la touche RETURN. Vous pourrez ainsi constater que le code de la touche DEL est 127 (\$7E).

Bien entendu, tout programme Basic, Pascal ou assembleur peut faire appel à ce caractère et tester le code. Ceci dit, nous ne voyons pas d'exemple spécial utilisant ce caractère, en dehors des possibilités graphiques que vous avez soulignées.

Dans la revue Pom's, je n'ai pu trouver le programme qui me permettrait de réaliser une "hard-copy" d'écran HGR, dessiné au moyen des instructions HPLLOT.

Jean-Claude Ricateau -
30400 Villeneuve les Avignon

A l'aide des instructions HPLLOT, vous pouvez dessiner sur la page graphique 1 ou 2. L'article "Les adresses du graphique" dans le Pom's 1 vous indique que la page 1 va de \$2000 à \$3FFF, et la page 2 de \$4000 à \$5FFF.

Ainsi, pour sauvegarder sur disquette la page 1 après y avoir dessiné avec des HPLLOTs, il suffit de faire BSAVE nomfic,A\$2000,L\$2000 où "nomfic" est le nom que vous voulez donner à l'image résultante.

Il ne vous reste plus qu'à utiliser un des nombreux programmes d'impression d'image, ou même une commande directe (telle que CTRL-I OH) d'impression d'écran graphique, si votre carte d'interface en possède une fonctionnant avec votre imprimante.

Suite à l'article "Donnez du caractère à votre imprimante, j'ai essayé l'impression en Multiplan sur mon imprimante, une Centronics 152. Pour passer en condensé, il faut faire ".*!T" en double condensé ".*!N" et en caractères normaux ".*!S". Sans le Pom's 4 (Les codes ASCII épiluchés), je n'aurais jamais trouvé !

Courrier des Lecteurs

Alexandre Duback

Hervé Guénec - 06950 Faucon

De la part des autres lecteurs qui auraient une Centronics 152, merci ...

Possédez-vous des informations comparatives entre le Pascal 1.1 et le Pascal 1.2 ? Quand la nouvelle version sera-t-elle disponible ?

B. Lambillon - 51 rue de l'Automne
- 1050 Bruxelles

La différence principale entre le Pascal 1.1 et le Pascal 1.2 tient à ce que ce dernier travaille en 128 Ko de mémoire centrale; en outre, plusieurs bugs du 1.1 ont disparu. Le produit devrait être sorti à la parution de ce numéro.

Comment peut-on avoir accès aux touches de l'Apple //e qui n'existaient pas sur l'Apple II Plus ? Le manuel est totalement muet sur ces nouvelles possibilités.

Dr. Denis Rousseaux - Auxon Dessus
- 25870 Geneuille

Les nouvelles touches ont les codes suivants en ASCII : 10 et 11 pour les flèches vers le bas et le haut, 127 pour DEL, 9 pour TAB. La pomme ouverte et la pomme pleine correspondent aux boutons 0 et 1 des manettes de jeu, comme vous pouvez le constater avec tout jeu utilisant les manettes. Les octets 49249 et 49250 permettent de tester l'état de ces touches (voir Pom's 13 p.45).

Apple, champion toutes catégories de l'incompatible !

En passant de l'Apple II+ à l'Apple //e, Apple a commis une erreur de conception. Si le progrès technologique justifiait le //e, le clavier, lui, a fait l'unanimité... des mécontents. Pour économiser deux bouts de chandelle et conserver l'ancien boîtier, on n'a pas hésité à plonger l'utilisateur dans un dédale de complexité dont la plupart ne se sortent pas. Deux utilisateurs sur trois rêvent d'un clavier normal pour remplacer le puzzle du //e, alors qu'il aurait été si simple de mettre la carte mère dans un boîtier séparé et de laisser l'acheteur choisir son type de clavier détachable.

Mais les choses n'en restent pas là, et Apple qui a fondé sa gloire sur l'adaptabilité vient de faire un bond en avant vers l'absurde en passant en un rien de temps de l'erreur à

l'aberration technique et commerciale.

Le //c devait initialement être destiné à une clientèle déjà équipée et qui attendait un portable avec impatience. Cela supposait donc une compatibilité totale. En fait, c'est d'INCOMPATIBILITE qu'il s'agit.

Si vous possédez 20.000 ou 30.000 francs de logiciel, vous pouvez les mettre à la poubelle. Si vous possédez une imprimante autre que l'Image Writer, vous pouvez la mettre à la poubelle. Si vous avez des drives en plus sur votre II+ ou //e, vous pouvez les mettre à la poubelle.

Votre joystick, quel que soit son modèle, prendra le même chemin et surtout ne comptez pas sur ProDOS pour vous aider, il est lui aussi spécial au //c.

Ni Multiplan, ni Apple Writer, ni Magicalc, ni PFS... bref plus aucun programme sérieux ne fonctionne. Il vous faudra tout racheter à moins qu'un sursaut de bon sens ne vous fasse tout simplement ignorer le //c, ce qui semble être la voie de la sagesse.

Voyons maintenant le Macintosh. Bien sûr l'incompatibilité est là aussi totale. Mais Apple, qui cherche toujours à se surpasser, a trouvé avec cette machine toute neuve un truc auquel les autres constructeurs n'avaient même pas pensé : rendre la machine incompatible avec elle-même; ça, c'est fort !

En effet, les Macs sont vendus en France avec un clavier Azerty et les programmes ne sont compatibles qu'avec un clavier Qwerty. Simple... mais il fallait y penser. Ainsi, les Multiplan vendus pour Mac ne tournent pas sur Mac...

Bien sûr, là encore votre chère imprimante ira à la casse; quant au drive supplémentaire, il vous faudra déboursier près d'un demi-million de centimes pour l'avoir, et tout ça pour pas tout à fait 400 malheureux Ks.

Le bloc numérique, vous le paierez en supplément, alors qu'il aurait dû être intégré à la souris, cette fameuse souris qui aurait dû être branchée sur le clavier pour éviter tous ces fils pendouillards et éviter en cas de panne d'être privé du clavier. Même aberration d'ailleurs pour le Duodisk dont toute panne de l'un entraîne la privation de l'autre et donc l'immobilisation de l'unité centrale.

Reste à savoir quelle merveille d'in-

cohérence nous réserve Apple avec le Lisa 2; parti comme ça l'est, nous ne serons sûrement pas déçus... Et, pendant ce temps-là, Toshiba nous fournit une machine avec deux drives de 700 Ko formatés chacun et un superbe écran graphique pour moins de 21.000 F TTC. Décidément, comme le dit la publicité, "l'empire contre-attaque" et risque fort de l'emporter si le bon sens n'est pas rapidement restauré.

Jean Chamoin - 13001 Marseille

Nous avons transmis votre lettre à Apple et publierons volontiers leur réponse, si vous pouvez nous l'envoyer. Ceci dit, je dois vous avouer que si, comme vous, j'ai trouvé le clavier du //e très désagréable durant la première demi-journée, je le préfère aujourd'hui cent fois à celui de l'IBM PC. Avez-vous essayé d'utiliser un tableur avec le PC? Comment obtient-on les chiffres avec un clavier Azerty sans se mettre en blocage majuscules? Au moins, avec le //e, je fais tout en clavier US, sauf le traitement de textes, et j'en suis ravi. De plus, quand des jeux américains utilisent pour le déplacement les touches I, J, K et M, cela devient fort peu pratique si l'on est forcé de travailler en Azerly. Enfin, pour utiliser un programme US sur le Macintosh, vous pouvez suivre la procédure indiquée dans le Pom's 13 (pages 66-67) ou, plus simplement mais moins définitivement, booter avec une disquette comportant un système français, l'éjecter, puis travailler alors avec le logiciel US.

1) J'ai écrit un programme en assembleur qui nécessite tant de mémoire libre que je dois le loger sur la carte langage. Or, ce programme utilise des routines de l'Applesoft et du Moniteur. Comment dois-je faire pour que tout marche?

2) J'aperçois sur la carte langage un circuit qui ressemble fortement aux ROMs de la carte mère. Qu'est-ce que c'est?

Monsieur Thierry Han - B.P. 301 - Yaoundé - Cameroun

1) La solution la plus simple est celle de l'appel indirect: il faut réserver une partie de la RAM normale (sous l'adresse \$BFFF) à une petite routine qui active et désactive la carte langage. Au lieu d'appeler directement le moniteur ou l'interpréteur Applesoft à partir du programme placé sur la carte langage, on appelle cette routine qui active la lecture de la ROM, appelle la routine demandée et enfin réactive la lecture (et éventuellement l'écriture) de la carte lan-

gage et rend la main au programme initial.

2) La carte langage comprend, outre les 16K d'extension RAM, la ROM du moniteur Autostart de l'Apple II Plus, qui avait remplacé l'Apple II. La ROM du moniteur située sur la carte mère de l'Apple n'est plus utilisée quand la carte langage est placée dans son slot.

Je ne parviens toujours pas à introduire, au sein d'un texte en Apple-Writer //e les commandes qui me permettraient d'imprimer avec des caractères autres que le Pica normal de mon Epson FX80/FT.

J'ai bien essayé d'utiliser le principe décrit dans le manuel, fondé sur Ctrl-V, par exemple avec PRINT CHR\$(15) pour obtenir les caractères condensés. Pouvez-vous me conseiller?

Roger Clavel - 34100 Montpellier

Pour envoyer des caractères à votre imprimante, je vous conseille, si ce n'est déjà fait, de lire "Donnez du caractère à votre imprimante", Pom's 9.

Il ne faut pas taper comme en Basic "PRINT CHR\$(15)" pour envoyer le caractère 15, mais faire Ctrl-V Ctrl-O Ctrl-V, car le Ctrl-O a 15 pour code ASCII. Voir à cet égard "Les codes ASCII épluchés", Pom's 4, qui indique ce qu'il faut taper pour engendrer tel ou tel code ASCII.

Enfin, le Courrier des Lecteurs du Pom's 10 (page 71) montre comment il faut traiter le cas particulier du code 0.

Je viens de prendre possession d'une ImageWriter avec une configuration complète //e. Comment fait-on pour copier son écran, en dehors des dessins préenregistrés de la disquette Toolkit fournie avec l'imprimante? Comment copier un dessin en double haute résolution, avec les 560*192 points de la carte Eve Chat Mauve?

Alain Dubois - 59230 St Amand les Eaux

En ce qui concerne les graphiques, Pom's 1, dans "les adresses du graphique", indique les zones mémoire des pages graphiques. Quand vous avez créé un graphique à partir d'un programme, il faut d'abord le sauvegarder sur disquette avec "BSAVE nom, A\$2000,L\$2000", ou A\$4000 s'il est en page 2. Il suffira alors de faire "BLOAD nom" pour charger la page en mémoire.

Votre programme Toolkit d'imprimante vous permet de charger puis d'imprimer une page graphique.

C'est tout ce qu'il vous reste à faire. Il n'existe pas à notre connaissance de programme de copie d'écran HGR double résolution. L'un de nos lecteurs en a peut-être réalisé un que nous pourrions publier...

En réponse au courrier de M. Januel (Pom's 12), voici la modification du programme MENU des disquettes Pom's qui permet l'impression sur une Epson avec interface Apple:

```
70 POKE 34,24:PRINT D1$*PR#"S
71 PRINT CHR$(0)
72 PRINT CHR$(27)"A"CHR$(10)
73 POKE 1657,80
75 CALL 37989
76 PRINT CHR$(27):CHR$(50)
77 PRINT D1$*PR#"0"
78 TEXT
```

Alain Meizoz

Avis aux lecteurs de Pom's ...

On nous déclare parfois que l'IBM PC est une bien meilleure machine pour communiquer que notre fidèle Apple II. Pour notre part, chaque fois que nous avons désiré relier notre Apple à un autre ordinateur, nous y sommes parvenus. Nous souhaitons réunir un catalogue des liaisons déjà établies entre un Apple et d'autres ordinateurs.

Nous prions donc nos lecteurs qui ont innové en ce sens de bien vouloir nous écrire pour nous dire à quels ordinateurs ils ont déjà relié leur Apple, et dans quel but. Bien entendu, les noms et adresses pourront rester confidentiels s'ils en expriment le souhait.

Vous pouvez indiquer en outre les problèmes que vous avez rencontrés, comment ils ont été résolus, combien coûte(nt) le(s) logiciel(s) éventuellement développé(s). Merci d'avance pour tous ceux qui pourront profiter de ces informations.

J'ai réalisé un logiciel pour la gestion et l'exécution de menuiseries à isolation renforcée: croisées, porte-fenêtre et châssis. Les personnes intéressées peuvent me contacter directement.

Christian de Grenier - 10, avenue de Verdun - 74100 Annemasse - Tél. (50) 37.54.49.

Erratum: Pom's 13, page 22, l'instruction 61130 comporte le terme INT(X/2) et non INT(X>2).

BRANCHÉS ?

MODEMS TELSAT

les transmissions ne d'ont pas au futur.



SATELCOM
INTERNATIONAL

69-71, rue du Chevaleret
75013 PARIS



54, rue de Dunkerque
75009 PARIS Tél.: 282.17.09
Métro: Gare du Nord (100 m)

SURPRIS ... LES PRIX!!!

PRIX T.T.C. ! POUR APPLE ET COMPATIBLES

Diskettes U.S 5" 1/4 SF/SD

Diskettes 3M 5" 1/4 SF/DD

Lecteur diskettes pour APPLE
(mécan. Japonais, entr. direct)
Carte synthétiseur de voix
Carte mémoire/langage 16 K Ram
Carte mémoire 128 K
Carte drive 13/16 sect.
Carte 80 colonnes
Carte imprimante parallèle
Carte imprimante + Buffer 32 K
Carte série
Carte super série
Carte Z80 - CP/M
Joystick de luxe 2 +, 2E, 2C

145 F/boîte 10 (exp. min. 5 boîtes Port 27 F)

175 F/boîte 10 (exp. min. 5 boîtes Port 27 F)

1950 F

(PORT 95 F)

350 F	Carte wilcard (déplombage)	600 F
450 F	Carte communication	600 F
1550 F	Carte IEEE - 488	1500 F
380 F	Carte copieur Eprom	700 F
740 F	Carte A/D - D/A 12 bit	1900 F
400 F	Carte horloge	550 F
1350 F	Carte musique	750 F
550 F	Carte RGB + Prise TV Secam	750 F
1050 F	Carte 6522 Via	600 F
650 F	Port pour une carte	27 F
180 F	Ventilo 10 W. super silencieux	350 F

Port Urgent ajouter 5,50 F

NOUVEAU: Pince spéciale pour diskettes (100 000 trous min.) (port: 13 F) **69 F**

Ordinateur multicompatible Forth, Basic, CP/M, MS-Dos, CP/M86

Écrivez, nous vous enverrons une liste plus complète de nos articles. **Revendeurs, contactez-nous.**

Notre devise: "DYNAMIT COMPUTER: MOINS CHER QUE MOI TU MEURS !"

pom's

		Montant TTC
• la disquette HAIFA Source	<input type="checkbox"/> au prix de 55 F la disquette (cf. Pom's n° 5)	
• le logiciel H-BASIC	<input type="checkbox"/> au prix de 150 F (cf. Pom's n° 8)	
• le logiciel MUSIC	<input type="checkbox"/> au prix de 80 F (cf. Pom's n° 10)	
• le Disk Manager	<input type="checkbox"/> au prix de 450 F (cf. Pom's n° 11)	
• DBSTAG (CP/M)	<input type="checkbox"/> au prix de 450 F (cf. Pom's n° 11)	
• Disquette de jeux A	<input type="checkbox"/> au prix de 80 F (cf. Pom's n° 12)	
• Disquette de jeux B	<input type="checkbox"/> au prix de 80 F (cf. Pom's n° 12)	
• le logiciel BASICIUM	<input type="checkbox"/> au prix de 150 F (cf. Pom's n° 13)	
• la disquette de démo sur 64K seulement	au prix de 55 F la disquette <input type="checkbox"/> CX Système <input type="checkbox"/> Jane	
• Recueil n° 1 de Pom's (n° 1 à 4)	<input type="checkbox"/> avec ses 3 disquettes au prix de 280 F <input type="checkbox"/> sans disquette au prix de 130 F <input type="checkbox"/> les 3 disquettes seules au prix de 150 F	
• Recueil n° 2 de Pom's (n° 5 à 8)	<input type="checkbox"/> avec ses 4 disquettes au prix de 320 F <input type="checkbox"/> sans disquette au prix de 130 F <input type="checkbox"/> les 4 disquettes seules au prix de 190 F	
Je désire recevoir :		
• les numéros de la revue Pom's	<input type="checkbox"/> 4 <input type="checkbox"/> 7 <input type="checkbox"/> 8 au prix de 35 F le numéro	
• les numéros de la revue Pom's	<input type="checkbox"/> 9 <input type="checkbox"/> 10 <input type="checkbox"/> 11 <input type="checkbox"/> 12 <input type="checkbox"/> 13 <input type="checkbox"/> 14 au prix de 40 F le numéro	
• les disquettes d'accompagnement des numéros	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/> 6 <input type="checkbox"/> 7 <input type="checkbox"/> 8 <input type="checkbox"/> 9 <input type="checkbox"/> 10 <input type="checkbox"/> 11 <input type="checkbox"/> 12 <input type="checkbox"/> 13 <input type="checkbox"/> 14 au prix de 55 F par disquette	
Je désire m'abonner pour 6 numéros à partir du n°		
<input type="checkbox"/> sans disquette au prix de 200 F		
<input type="checkbox"/> avec disquettes au prix de 480 F		
TOTAL :		

Envoyez ce bon de commande et votre règlement à :

Éditions MEV — 64-70, rue des Chantiers — 78000 Versailles

Nom _____

Adresse _____

Ces tarifs comprennent l'envoi postal en France Métropolitaine, CEE et Suisse (voie aérienne exceptée)

Supplément avion : 10 F par numéro et / ou disquette

FIABILITE
PERFORMANCE

Datalife

DE Verbatim®

3 $\frac{1}{2}$ 5 $\frac{1}{4}$ 8"



*La satisfaction
garantie*

10 Minidis



SECURITE

LONGEVITE

QUALITE

- Disquette de nettoyage
- Analyseur d'unité de disque
- Cassettes et mini-cassettes digitales certifiées

La gamme référence

plus

l'environnement de votre système



- Rubans imprimant cassette et bobine PEGASUS
- Rangement des disquettes
- Filtres écran SUN ETN
- Produits entretien COMPU CLEAN
- Chargeurs de disquettes + Roues d'impression
- Tapis antistatiques

EGLE 793 11 47

BFI Electronique

Tél : (1) 533.01.37
9, rue Yvart
75013 PARIS

Faites la connaissance de vos nouveaux collaborateurs: Macintosh et Davong



Macintosh™, le dernier-né de chez Apple, est en train de révolutionner l'informatique individuelle comme l'ont fait ses ancêtres, les premiers Apple.

Associé aux périphériques Davong, le Macintosh allie la facilité d'emploi à la puissance, la souplesse et de nombreuses possibilités d'extension au service de vos affaires.

Le nouveau système Davong Mac Disk, que vous pouvez vous procurer auprès des concessionnaires Davong dans le monde entier, vous offre de 10 à 32 millions d'octets de mémoire à disques. Grâce à Mac Disk, vous obtenez enfin la capacité augmentée et la fonctionnalité améliorée qu'il vous faut pour résoudre vos problèmes de gestion. Le tout à un prix très raisonnable, sans affecter la vitesse et la simplicité d'emploi qui caractérisent le Macintosh.

Une autre bonne nouvelle de chez Davong : l'arrivée très prochaine de son système de sauvegarde à bande en continu d'une capacité de 28 millions d'octets, qui assure la sauvegarde rapide de vos fichiers en toute sécurité, ainsi qu'un système multi-tâche de gestion de réseau de zone locale qui permet à tous les Macintosh de votre entreprise de partager des données, des programmes et de bénéficier en commun des avantages du Mac Disk.

Pour obtenir de plus amples informations, veuillez vous adresser à votre concessionnaire Davong.



importateur :
HIT MICRO SYSTEMES
171, avenue Charles-de-Gaulle
92200 NEULLY Tél. : (1) 738.28.80

Apple et Macintosh sont des marques de Apple Computer Inc.